

Copyright ©1995–2005 Tobias Oetiker and Contributors. All rights reserved.
Copyright © 1999, 2007 for the Polish translation and extension JG, RK and TP
All rights reserved.

This document is free; you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or (at your option) any later version.

This document is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with this document; if not, write to the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Copyright © 1995–2005 Tobias Oetiker i wszyscy współautorzy *Wprowadzenia*.
Copyright © 1998, 2007 polskiego tłumaczenia i opracowania JG, RK i TP.

Niniejszy dokument jest wolno dostępny; można go rozpowszechniać i/lub zmieniać zgodnie z postanowieniami Ogólnej Licencji Publicznej GNU – takiej, jaką opublikowała fundacja Free Software Foundation w wersji 2 tejże Licencji, albo (Wasz wybór) w dowolnej późniejszej.

Dokument jest rozpowszechniany w nadziei, że będzie użyteczny, jednakże BEZ ŻADNEJ GWARANCJI, nawet bez jakiegokolwiek domyślnej gwarancji WYNIKAJĄCEJ Z NABYCIA lub ODPOWIADANIA KONKRETNEMU CELOWI. Więcej szczegółów znajdziecie w Ogólnej Licencji Publicznej GNU.

Do dokumentu powinna być dołączona kopia Ogólnej Licencji Publicznej GNU; jeśli jej nie ma, to napiszcie do: the Free Software Foundation, Inc., 675 Mass Ave, Cambridge, MA 02139, USA.

Najnowszą polskojęzyczna wersja *Wprowadzenia* znajduje się pod adresem:
<http://www.ctan.org/tex-archive/info/lshort/polish/>

Najnowsza anglojęzyczna wersja *Wprowadzenia* znajduje się w katalogu:
<http://www.ctan.org/tex-archive/info/lshort>

Richard Nagy, Philipp Nagele, Lenimar Nunes de Andrade, Manuel Oetiker, Urs Oswald, Martin Pfister, Demerson Andre Polli, Nikos Pothitos, Maksym Polyakov Hubert Partl, John Reffing, Mike Ressler, Brian Ripley, Young U. Ryu, Bernd Rosenlecher, Chris Rowley, Risto Saarelma, Hanspeter Schmid, Craig Schlenker, Gilles Schintgen, Baron Schwartz, Christopher Sawtell, Miles Spielberg, Geoffrey Swindale, Laszlo Szathmary, Boris Tobotras, Josef Tkadlec, Scott Veirs, Didier Verna, Fabian Wernli, Carl-Gustav Werner, David Woodhouse, Chris York, Fritz Zaucker, Rick Zaccone, oraz Mikhail Zotov.

Od autorów polskiej wersji

Za pomoc redakcyjną dziękujemy Staszкови Wawrykiewiczowi i Włodkowi Macewiczowi.

tabular, 38, 39, 53, 113, 116, 117, 119–121	\title, 33, 83	\vec, 48
tabular*, 116–118	\titlegraphic, 83	\vector, 87
tabularx, 118	<i>titlepage</i> , 10	\verb, 38, 43, 99
tabularx, 116, 118	Tkadlec Josef, 58	verbatim, 38, 72
tap, 122	.toc (plik), 13, 32	verbatiminput, 72
\temporal, 83	\today, 21	verse, 37
.tex (plik), 8, 12	\topmargin, 112	\vspace, 110
\TeX, 21	\totalheight, 114, 115	\vspace*, 110
<u>TeXnicCenter</u> , 75	tryb matematyczny, 45	w-tył-ciach, 5
\texorpdfstring, 79, 80	<i>twocolumn</i> , 10, 18	wektor, 48
\text, 55	<i>twoside</i> , 10	\widehat, 48
\textbf, 103, 105	txfonts, 106	\widetilde, 48
\textcelsius, 23	tylda (~), 31	\width, 114, 115
textcomp, 23	ułamek	wielokropek, 24, 51
\texteuro, 23	▷ piętro, 49	Williams Graham, 11
\textheight, 112	▷ zwykły, 49	Woliński Marcin, 29, 30
\textit, 103	umlaut, 25	wstawka, 40
\textmd, 103	\uncover, 83	www, 23
\textnormal, 103	\underbrace, 48	wzorce podziału, 25
\textrm, 55, 103	underfull hbox, 19	X ₂ , 28
\textsc, 103	\underline, 48	XeTeX, 28
\textsf, 103	<i>unicode</i> , 80	<u>xfig</u> , 65
\textsl, 103	Unicode, 8, 26, 28, 29, 77, 80	<u>xpdf</u> , 74
\textstyle, 55	\unitlength, 85–87	xy, 95
\texttt, 103	\updownarrow, 50	\xymatrix, 95
\textup, 103	url, 23	<u>yap</u> , 9
\textwidth, 68, 112	\url, 99	zalety L ^A T _E Xa, 3
Thành Hàn Thế, 75	URL, 23	Zapf Hermann, 106
thebibliography, 68	\usebox, 91	znak
\thicklines, 87, 90, 92	\usepackage, 6, 11, 12, 23, 26, 67, 77, 101, 122	▷ całki, 50
\thinlines, 87, 90, 92	<i>utf8</i> , 8, 26	▷ sumowania, 50
\thispagestyle, 11	\vdots, 51	żywa pagina, 11, 71
tilda, 23		
\tiny, 103		

L^AT_EX jest dostępny na większości platform sprzętowych, począwszy od PC i Macintosh, a skończywszy na dużych systemach wyposażonych w system UNIX czy VMS. W wielu sieciach uniwersyteckich można spotkać gotowe do użytku instalacje L^AT_EXa. Informacje, jak rozpocząć pracę w lokalnej instalacji L^AT_EXa, można znaleźć w [16]. Jeżeli nie wiesz, jak zacząć pracę z L^AT_EXem, to zapytaj się osoby, od której otrzymałeś niniejsze *Wprowadzenie*. W tym dokumencie nie poruszamy spraw związanych z instalowaniem i konfigurowaniem systemu L^AT_EX. Mówimy wyłącznie o tym, jak pisać dokumenty, aby mogły być przetwarzane przez L^AT_EXa.

Gdybyś potrzebował jakichkolwiek materiałów dotyczących L^AT_EXa, to zajrzyj do jednego z archiwów sieci CTAN. Główny węzeł CTAN ma adres <http://www.ctan.org>. Można też korzystać z innych węzłów, takich jak <http://www.dante.de> lub <ftp://ftp.dante.de> albo dowolnego z wielu archiwów lustrzanych, na przykład polskiego <ftp://ftp.gust.org.pl>.

Wszystkie zasoby w każdym archiwum CTAN znajdują się w katalogu `tex-archive`. W dalszej części książki będziemy wielokrotnie używać odsyłaczy do zasobów CTANu, ale będziemy w nich pomijać protokół, adres konkretnego węzła oraz początkowy katalog `tex-archive`, to znaczy będziemy używać zapisu `CTAN://ścieżka-do-zasobu` zamiast <http://www.ctan.org/tex-archive/ścieżka-do-zasobu>.

W katalogu `CTAN://systems` znajdziesz oprogramowanie niezbędne do uruchomienia L^AT_EXa na twoim komputerze.

Od autorów polskiej wersji

Ponieważ uznaliśmy, że niektóre tematy autorzy *Wprowadzenia* przedstawili zbyt lakonicznie, zdecydowaliśmy się na ich poszerzenie. Zmiany w stosunku do oryginału dotyczą w szczególności: zagadnień przygotowywania dokumentów w języku polskim (punkt 2.5 i wiele uzupełnień w innych punktach), opisu klasy `letter` (punkt 2.14) oraz informacji dotyczących tworzenia plików PDF (punkt 4.7). Ponieważ w naszej opinii w oryginale *Wprowadzenia* zdecydowanie za mało miejsca zajmuje ważne zagadnienie składania tabel, dodaliśmy poświęcony w całości temu tematowi obszerny punkt 6.8.

Jeżeli masz pomysł, co należałoby dodać, usunąć lub zmienić w tym dokumencie bądź jego tłumaczeniu – napisz. Jesteśmy szczególnie zainteresowani opiniami początkujących użytkowników L^AT_EXa o tym, które fragmenty trudno zrozumieć i wymagają lepszego przedstawienia.

Ryszard Kubiak, R.Kubiak@guests.ipipan.gda.pl

Tomasz Przechlewski, T.Przechlewski@GUST.org.pl

Z autorem oryginalnej wersji angielskiej możesz się skontaktować, pisząc pod adresem: Tobias Oetiker (oetiker@ee.ethz.ch).

Polecamy stronę <http://www.gust.org.pl> *Polskiej Grupy Użytkowników Systemu T_EX GUST* jako dodatkowe, bogate źródło informacji o T_EXu.

J („j” bez kropki), 25
 Jackowski Bogusław, 20, 27, 84, 122
 jednostki miary, 109
 jpeg2ps, 66
 .jpg (plik), 67, 76, 80
 kaszta, 102
 Kew Jonathan, 28
 Kile, 75
 klasa
 ▷ article, 9
 ▷ book, 10
 ▷ letter, 10
 ▷ mwart, 30
 ▷ mwbook, 30
 ▷ mwrep, 30
 ▷ report, 10
 ▷ slides, 9
 klej, 113
 Knuth Donald E., 1, 85
 kodowanie
 ▷ LGR, 28
 ▷ OT1, 27
 ▷ OT4, 27
 ▷ T1, 27
 ▷ T2A, 28
 ▷ T2B, 28
 ▷ T2C, 28
 ▷ X2, 28
 Kołodziejska Hanna, 20
 komentarz, 6
 kropka, 24
 \label, 34, 42, 46, 68
 Lamport Leslie, 2
 \large, 103
 \Large, 103
 \LARGE, 103
 \LaTeX, 21
 L^AT_EX2.09, 2
 L^AT_EX 2_ε, 2
 L^AT_EX3, 2, 4
 \LaTeXe, 21
 latexsym, 12
 latin2, 8, 26
 layout, 111
 \ldots, 24, 51
 \left, 50, 51
 \leftmark, 72
 legalpaper, 10
 \leq, 59
 legno, 10
 letter, 44
 letterpaper, 10, 81
 LGR, 28
 ligatura, zob. spójka
 \line, 87, 92
 \linebreak, 17, 18
 \linespread, 107
 \linethickness, 90, 92
 linia podstawowa, 51, 114, 116
 \listoffigures, 42
 \listoftables, 42
 .lof (plik), 13, 32
 .log (plik), 13, 76
 longtable, 119
 longtable, 119
 .lot (plik), 13, 32
 Macewicz Włodzimierz, 11, 70
 \mainmatter, 33, 79
 \makebox, 114, 115
 makeidx, 12, 69
 \makeindex, 69
 \makeindex, 13, 69, 70
 \maketitle, 33
 \maktexsr, 73
 \marginparpush, 112
 \marginparsep, 112
 \marginparwidth, 112
 Marszałkowska Joanna, 123
 marvosym, 24
 math, 45
 \mathbb, 47
 \mathbf, 57, 105
 mathcal, 64
 \mathcal, 105
 \mathit, 105
 \mathnormal, 105
 \mathrm, 55, 105
 mathscr, 64
 \mathsf, 105
 \mathtt, 105
 \mbox, 20, 24, 114
 \medskip, 110
 METAPOST, 76, 77
 MeX, 30
 minipage, 114
 Mittelbach Frank, 2
 modulo, 49
 Morison Stanley, 106
 .mps (plik), 76, 80
 MS Office, 66
 MS Visio, 66
 \multicolumn, 39
 \multirow, 86, 89
 multiline, 54
 nawias, 50
 \newcommand, 99
 \newenvironment, 99, 100
 \newline, 17, 18
 \newpage, 17, 18
 \newsavebox, 91
 \newtheorem, 56
 \newtheoremstyle, 56
 \noindent, 108
 \nolinebreak, 17
 \nomathsymbols, 30
 \nonumber, 54
 \nopagebreak, 17
 \normalsize, 103
 \not, 59
 \notitlepage, 10
 Nowacki Janusz, 27
 obracanie
 ▷ rysunku, 68
 ▷ tabeli, 68
 \oddsidemargin, 112
 odstęp, 4
 ▷ na początku wiersza, 4
 ▷ po instrukcji, 5
 ▷ podwójny, 107
 ▷ poziomy, 108
 ▷ w trybie matematycznym, 51
 α, 25
 ogranicznik, 50
 Olko Mariusz, 29
 onecolumn, 10
 oneside, 10
 \only, 83
 Oostrum Piet van, 71
 opcje, 10
 openany, 10, 18
 \opening, 44
 openright, 10, 18

2.4.1.	Cudzysłowy	21
2.4.2.	Pauzy i myślniki	21
2.4.3.	Odstępy niełamliwe	22
2.4.4.	Tylda (\sim)	23
2.4.5.	Oznaczenie stopni (\circ)	23
2.4.6.	Symbol waluty euro (€)	23
2.4.7.	Wielokropek (...)	24
2.4.8.	Ligatury	24
2.4.9.	Akcenty i znaki specjalne	25
2.5.	L ^A T _E X wielojęzyczny	25
2.5.1.	Język polski w dokumentach	28
2.6.	Odstępy między wyrazami	30
2.7.	Tytuły, śródtytuły i punkty	31
2.8.	Odsyłacze	34
2.9.	Przypisy	34
2.10.	Wyróżnienia	35
2.11.	Otoczenia	35
2.11.1.	Otoczenia <code>itemize</code> , <code>enumerate</code> i <code>description</code>	36
2.11.2.	Otoczenia <code>flushleft</code> , <code>flushright</code> i <code>center</code>	36
2.11.3.	Otoczenia <code>quote</code> , <code>quotation</code> i <code>verse</code>	37
2.11.4.	Streszczenie	37
2.11.5.	Symulacja maszynopisu	38
2.11.6.	Otoczenie <code>tabular</code>	38
2.12.	Wstawki	40
2.13.	Ochrona poleceń kruchych	43
2.14.	Listy	43
3.	Wyrażenia matematyczne	45
3.1.	Wstęp	45
3.2.	Grupowanie	47
3.3.	Części składowe wyrażen matematycznych	47
3.4.	Odstępy w trybie matematycznym	51
3.5.	Wyrównywanie w pionie	52
3.6.	Fantomy	54
3.7.	Stopień pisma	55
3.8.	Twierdzenia, definicje, itp.	56
3.9.	Symbole półgrube	57
3.10.	Zestawienie symboli matematycznych	58
4.	Rysunki, skorowidze, generowanie plików PDF...	65
4.1.	Włączanie grafiki w formacie EPS	65
4.2.	Spis literatury	68
4.3.	Skorowidze	69
4.4.	Paginy górne i dolne	71

Skorowidz

Uwaga: hasła wyróżnione imitacją pisma maszynowego, oznaczają polecenia (jeżeli są poprzedzone znakiem w-tył-ciacha) lub otoczenia; wartości opcji klas i pakietów oznaczono *odmianą pochytą* imitacji pisma maszynowego; programy zostały wyróżnione podkreśleniem zaś hasła złożone krojem bezszeryfowym oznaczają pakiety.

$\!$, 52	<u>Acrobat Reader</u> , 74, 76, 81	beamer, 9, 81–83
$\$, 45$	acute, 25	.bib (plik), 69
$\', 28$	<code>\addcontentsline</code> , 32	<code>\bibitem</code> , 68
$\', 48$	<code>\address</code> , 44	<code>\big</code> , 51
$\langle, 45$	<code>\addtolength</code> , 111	<code>\Big</code> , 51
$\backslash, 45$	æ , 25	<code>\bigg</code> , 51
$\backslash, 46, 51$	akcenty, 25	<code>\Bigg</code> , 51
$\backslash-, 20$	\triangleright matematyczne, 48	<code>\biggl</code> , 55
..., zob. wielokropek	<code>align</code> , 54	<code>\biggr</code> , 55
$\backslash:, 51$	<i>all</i> , 94	<code>\bigskip</code> , 110
$\backslash;, 51$	<code>\alt</code> , 83	<code>\binom</code> , 49
$\backslash@, 31$	<code>ambsy</code> , 57	block, 83
$\backslash[, 45$	<code>amsmath</code> , 47, 64	<code>\bmod</code> , 49
$\%, 6$	<code>amsmath</code> , 46, 49–52, 54, 57	<code>\boldmath</code> , 57
Bib _T E _X , 69	<code>amssymb</code> , 47, 58, 64	<code>\boldsymbol</code> , 57
$\backslash\backslash, 17, 36, 37, 39, 43, 110$	<code>amsthm</code> , 56, 57	Braams Johannes, 25
$\backslash\backslash*, 17$	<code>\and</code> , 33	.bst (plik), 69
wysiwyg, 2, 3	<code>\appendix</code> , 31, 32	calc, 111
$\backslash], 45$	<code>\ar</code> , 95	<code>\caption</code> , 41–43
$\backslash\sim, 47$	argument, 6	Carlisle David, 58, 65
$\backslash\sim-, 47$	\triangleright opcjonalny, 6	Casartelli Fabio, 120
$\backslash , 39$	array, 120, 121	<code>\cc</code> , 44
$\backslash\sim, 31$	array, 52, 53	<code>\cdots</code> , 51
<i>10pt</i> , 10	<code>arraycolsep</code> , 53	center, 36, 68
<i>11pt</i> , 10, 51	<code>\atop</code> , 49	<code>\centering</code> , 68
<i>12pt</i> , 10, 51	<code>\author</code> , 33, 78, 83	<code>\chapter</code> , 31, 72
	.aux (plik), 13, 32, 34, 69	<code>\chaptermark</code> , 72
$\backslash\text{a}'\text{o}$, 28	<i>b5paper</i> , 10	<code>\choose</code> , 49
<i>a4paper</i> , 10, 81	babel, 11, 25–29	<code>\circle</code> , 88
<i>a5paper</i> , 10	<code>\backmatter</code> , 33	<code>\circle*</code> , 88
abstract, 37	backslash, 5	<code>\cite</code> , 68
<u>Acrobat</u> , 78, 79	<code>\baselineskip</code> , 107	<code>\cleardoublepage</code> , 18, 42
<u>Acrobat Distiller</u> , 74	.bbl (plik), 69	<code>\clearpage</code> , 18, 42

6.3.4. Odstępy pionowe	110
6.4. Układ graficzny strony	111
6.5. Więcej o odległościach	113
6.6. Pudełka	113
6.7. Kreski i podpory	115
6.8. Więcej o składaniu tabel	116
6.8.1. Tabele o zadanej szerokości	116
6.8.2. Pakiet <code>longtable</code>	119
6.8.3. Pakiet <code>array</code>	120
6.8.4. Pakiet <code>tap</code>	122
Bibliografia	123
Skorowidz	125

Bibliografia

- [1] Borzyszkowski Andrzej: Bib \TeX – narzędzie do przygotowania bibliografii. Biuletyn GUST 1999 (13), Dostępny także w <http://www.ipipan.gda.pl/~andrzej/papers/bibtex.pdf>
- [2] Carlisle David P.: *Packages in the 'graphics' bundle*. Dokument dostępny w zestawie pakietów „graphics” w pliku `grfguide.tex`.
- [3] Chwałowski Robert: *Typografia typowej książki*, Helion 2001, ISBN: 83-7197-545-7, Por. też <http://www.typografia.ogme.pl/>.
- [4] Diller Antoni: *L^AT_EX wiersz po wierszu*, tłum. Jan Jełowicki, Helion, Gliwice 2001, ISBN: 83-7197-341-1.
- [5] Eijkhout Victor: *T_EX by Topic, A T_EXnician's Reference*, Addison-Wesley, Wokingham, England 1992 ISBN: 0-201-56882-9 Dostępny w <http://www.eijkhout.net/tbt/>.
- [6] Mittelbach Frank i inni: *L^AT_EX Companion*, 2nd Edition, Addison-Wesley, Reading 2004, ISBN: 0201362996.
- [7] Goossens Michel, Rahtz Sebastian, Mittelbach Frank: *The L^AT_EX Graphics Companion*. Addison-Wesley, Reading, Massachusetts, 1997, ISBN 0-201-85469-4.
- [8] Hobby John D.: *A User's Manual for MetaPost*. Dostępny w <http://cm.bell-labs.com/who/hobby/>. Polskie tłumaczenie Joanny Marszałkowskiej jest dostępne w <ftp://ftp.gust.org.pl/pub/GUST/doc/mpint-pl.pdf>.
- [9] Hoenig Alan: *T_EX Unbound*. Oxford University Press, 1998, ISBN0-19-509685-1+; 0-19-509686-X.
- [10] Jackowski Bogusław: *Co ma Bézier do B-spline'a?* Biuletyn GUST 2001 (17), ISSN: 1230-5650, Dostępny także w <ftp://ftp.gust.org.pl/pub/GUST/bulletin/17/jacko01.ps.gz>.
- [11] Knuth Donald E.: *The T_EXbook*, Addison-Wesley Publishing Company 1984, ISBN 0-201-13448-9.
- [12] L^Ampport Leslie: *L^AT_EX: A Document Preparation System*. 2nd ed., Addison-Wesley, Reading 1994, ISBN 0-201-52983-1. Polskie tłumaczenie Marii Wolińskiej i Marcina Wolińskiego *L^AT_EX System opracowywania dokumentów. Podręcznik i przewodnik użytkownika*, WNT, Warszawa 2004, ISBN 83-204-2878-5.
- [13] L^AT_EX3 Project Team: *L^AT_EX 2_ε for authors*. Dokument dostępny w pliku `usrguide.tex` w dystrybucji L^AT_EX 2_ε.

Spis tabel

1.1. Wybrane pakiety z podstawowej dystrybucji L ^A T _E Xa	12
2.1. Torba pełna symboli euro	24
2.2. Akcenty i znaki specjalne	25
2.3. Opcjonalny argument otoczeń <code>table</code> i <code>figure</code>	41
3.1. Akcenty matematyczne	58
3.2. Litery alfabetu greckiego	58
3.3. Symbole relacji	59
3.4. Symbole operacji dwuargumentowych	60
3.5. Symbole zmiennej wielkości	60
3.6. Strzałki	60
3.7. Ograniczniki	61
3.8. Duże ograniczniki	61
3.9. Różne symbole	61
3.10. Symbole niematematyczne	61
3.11. Ograniczniki (pakiet AMS)	61
3.12. Symbole Greckie i Hebrajskie (pakiet AMS)	61
3.13. Symbole relacji (pakiet AMS)	62
3.14. Strzałki (pakiet AMS)	62
3.15. Negacje symbolów relacji i strzałek (pakiet AMS)	63
3.16. Relacje dwuargumentowe (pakiet AMS)	63
3.17. Różne symbole (pakiet AMS)	64
3.18. Kroje pisma dostępne w trybie matematycznym	64
4.1. Znaczenie ważniejszych kluczy polecenia <code>\includegraphics</code>	66
4.2. Przykłady składni polecenia <code>\index</code>	70
6.1. Polecenia wyboru krojów i odmian	103
6.2. Polecenia jednoczesnego wyboru stopnia pisma i interlinii	103
6.3. Wielkość stopnia pisma w klasach standardowych	104
6.4. Polecenia wyboru fontów w trybie matematycznym	105
6.5. L ^A T _E Xowe jednostki miary	109
6.6. Tytuł tabeli	119

Złożone tabele wyglądają następująco:

x y z	x y z	1 1 1	x y z			x y z		
x y z	x y z	1 1	x y z	x y z	2 2 2	x y z		
x y z	x y y		x y z	x y z	2 2	x y z	x y z	
x y z			x y z	x y y		x y z	x y z	3 3 3
x y z			x y z			x y z	x y y	3 3

Kolejnym użytecznym rozszerzeniem zdefiniowanym w `array` jest możliwość bardziej szczegółowego określenia sposobu formatowania rubryk tabeli, niż ma to miejsce w standardowym otoczeniu `tabular`, gdzie w zasadzie możemy jedynie określić sposób justowania zawartości rubryk w poszczególnych kolumnach. Wracając do przykładu tabeli zawierającej różne „parametry” dotyczące przełęczyci i wniesień, założmy, że nazwę wniesienia chcemy złożyć kursywą, a długość – odmianą półgrubą. Można to oczywiście osiągnąć, wstawiając odpowiednie polecenia do każdej rubryki danej kolumny, ale sprawniej będzie zastosować zdefiniowane w pakiecie `array` konstrukcje:

`>{polecenia}` lub `<{polecenia}`

Konstrukcji `>{polecenia}` można użyć w preambule *przed* `c`, `l`, `r`, `p` oraz `m` i `b`. Jej działanie polega na wstawieniu *polecen* na początku każdej rubryki w tej kolumnie. Podobnie, `<{polecenia}` można użyć *po* `c`, `l`, `r`, `p`, `m` i `b`. W rezultacie *polecenia* zostaną wstawione na końcu każdej rubryki w tej kolumnie.

Wracając do przykładu, oto preambula tabeli wykorzystująca omawiane konstrukcje pakietu `array`:

```
\begin{tabular}{|>{\itshape}l|r|>{\bfseries}r|r|}\hline
Nazwa & \multicolumn{2}{|c|}{Wys. w m~n.p.m.} & & & \\
Długość & \multicolumn{2}{|c|}{Nachylenie \%}\backslash \\
...
```

Wynik jest następujący:

Nazwa	Wys. w m n.p.m.		Długość w km	Nachylenie %	
	początek	koniec		śr.	max
<i>Col du Galibier</i>	1401	2646	18,1	6,9%	14,5
<i>Alpe D'Huez</i>	724	1815	13,5	7,7%	15,0
<i>Passo Gavia</i>	1734	2618	13,5	6,5%	20,0

Do zmiany głębokości konkretnego wiersza służy opcjonalny argument polecenia `\`, np. `\[2pt]`. Wysokość i głębokość wiersza można zmienić za pomocą niewidzialnej kreski, np. `\rule[-3mm]{0mm}{8mm}`. Przykładowo, w powyższym przykładzie tabela będzie wyglądała lepiej, jeżeli dodamy do pierwszego wiersza:

```
Col du Galibier\rule[-3.5pt]{0pt}{15pt}
```

Spowoduje to wstawienie dodatkowego odstępu między kreską nad pierwszym wierszem tabeli a zawartością wiersza.

typograficznym. Pierwszą wersję \LaTeX a opracował Leslie Lamport [12]. Do formatowania dokumentu \LaTeX używa programu \TeX . Pielęgnacją dzisiejszych wersji \LaTeX a zajmuje się Frank Mittelbach.

Kilka lat temu pakiet \LaTeX został rozszerzony przez zespół o nazwie $\text{\LaTeX}3$ (*$\text{\LaTeX}3$ team*), kierowany przez Franka Mittelbacha. Celem tego rozszerzenia było wprowadzenie kilku od dawna postulowanych ulepszeń oraz unifikacja rozmaitych odmian \LaTeX a, które rozpowszechniły się od chwili powstania kilkanaście lat wcześniej \LaTeX a w wersji 2.09. Nową wersję pakietu nazwano $\text{\LaTeX}2\epsilon$, w celu odróżnienia jej od wersji poprzednich. Niniejszy dokument opisuje właśnie $\text{\LaTeX}2\epsilon$.

Słowo \LaTeX należy wymawiać „lej-tech” albo „la-tech.” Jeżeli nie można zapisać symbolu \LaTeX , to zamiennie należy użyć zapisu \LaTeX . $\text{\LaTeX}2\epsilon$ wymawiamy „la-tech dwa i”, a zamienną wersją zapisu jest $\text{\LaTeX}2\epsilon$.

1.2. Podstawy

1.2.1. Autor, redaktor i zecer

Aby wydać książkę, autor dostarcza maszynopis do wydawnictwa. W wydawnictwie redaktor decyduje o układzie graficznym dokumentu (szerokość szpalty, krój pisma, odstępy przed i po tytułach rozdziałów itd.). Redaktor zapisuje swoje decyzje w maszynopisie, w formie odpowiednich instrukcji, i przekazuje go zecerowi. Na podstawie maszynopisu oraz instrukcji zecer wykonuje skład.

Redaktor-człowiek odgaduje, co miał na myśli autor, gdy zapisywał maszynopis. Wykorzystując swoje doświadczenie zawodowe, ustala, które miejsca w maszynopisie oznaczają tytuły rozdziałów, podrozdziałów, cytaty, przykłady, wzory matematyczne itd.

\LaTeX gra rolę redaktora, a \TeX – zecera. Z tym że \LaTeX jest „zaledwie” programem komputerowym i dlatego potrzebuje dodatkowej pomocy autora, który powinien dostarczyć niezbędnych do składu informacji o strukturze logicznej dokumentu. Informacje te autor zapisuje w pliku źródłowym dokumentu jako „polecenia dla \LaTeX a”.

Praca z \LaTeX em zdecydowanie różni się od podejścia stosowanego w procesorach tekstu typu WYSIWYG¹, takich jak MS Word albo OpenOffice. Pierwszy sposób można określić „formatowaniem logicznym” drugi – „formatowaniem wizualnym”. Używając programów typu WYSIWYG, autor decyduje interaktywnie o wyglądzie graficznym dokumentu, w miarę dopisywania tekstu. Przez cały czas widzi na ekranie, jak tekst będzie wyglądał po wydrukowaniu.

Używając \LaTeX a, nie można na ogół oglądać dokumentu w jego ostatecznej postaci i jednocześnie dopisywać tekstu. Można natomiast obejrzeć

¹ang. *What you see is what you get* (dostaniesz to, co widzisz).

Tabele złożone z użyciem otoczenia `tabular` nie są automatycznie dzielone między stronami. Do składu tabel, które nie mieszczą się na pojedynczej stronie, służy pakiet `longtable`, opisany w następnym punkcie.

6.8.2. Pakiet longtable

Pakiet definiuje otoczenie `longtable`, pozwalające składać tabele ciągnące się przez wiele kolejnych stron dokumentu. Aby zapewnić jednakowe szerokości rubryk na wszystkich stronach, wymagana jest dwukrotna kompilacja dokumentu. Oto przykład:

```
\begin{longtable}{|l|r|r|r|}
\caption{Tytuł tabeli}\hline
\multicolumn{4}{|c|}{To jest nagłówek pierwszej strony}\hline
nazwa & wysokość & długość & nachylenie \hline
k-1 & k-2 & k-3 & k-4 \hline
\endfirsthead
\hline
\multicolumn{4}{|c|}{To jest nagłówek następnych stron}\hline
\hline k-1 & k-2 & k-3 & k-4\hline
\endhead
\hline \multicolumn{4}{|c|}{Stopka tabeli}\hline
\endfoot
\hline \multicolumn{4}{|c|}{Stopka na ostatniej stronie}\hline
\hline
\endlastfoot
Col du T\`el'egraphe & 1,566 & 12,0 & 6,7\% \hline
Col du Galibier & 2,646 & 18,1 & 6,9\% \hline
Col de la Madeleine & 2,000 & 25,4 & 6,1\% \hline
%% ... 17 pominiętych wierszy ...
La Bola del Mundo & 2257 & 21,8 & 6,2\% \hline
\end{longtable}
```

Po złożeniu efekt jest następujący¹⁰:

Tabela 6.6: Tytuł tabeli

To jest nagłówek pierwszej strony			
nazwa	wysokość	długość	nachylenie
k-1	k-2	k-3	k-4
Col du Télégraphe	1,566	12,0	6,7%
Col du Galibier	2,646	18,1	6,9%
Col de la Madeleine	2,000	25,4	6,1%
Stopka tabeli			

¹⁰Dane z tabeli pochodzą z katalogu <http://www.salite.ch>.

- Istnieje wiele bezpłatnych pakietów poszerzających typograficzne możliwości \LaTeX a. Dostępne są na przykład pakiety umożliwiające wstawianie do dokumentów grafiki w formacie Postscript, tworzenie hipertekstowej wersji dokumentów w formacie PDF czy też przygotowanie spisów bibliograficznych według ściśle określonych reguł, obowiązujących w różnych wydawnictwach. Opis wielu z tych pakietów można znaleźć w podręczniku [6].
- \LaTeX zachęca autorów do tworzenia dokumentów o dobrze określonej strukturze.
- \TeX – program formatujący używany przez $\LaTeX 2_{\epsilon}$ – jest bezpłatny i w najwyższym stopniu przenośny. Dzięki temu można działać na praktycznie dowolnej platformie systemowo-sprzętowej.

\LaTeX ma także pewne wady, chociaż ciężko mi znaleźć jakąkolwiek istotną. Jestem jednak pewien, że inne osoby wskażą ci ich setki;-)

- \LaTeX nie działa u tych, którzy zaprzędali swoje dusze. . .
- Chociaż przez zmianę niektórych parametrów można dostosowywać predefiniowane układy graficzne do własnych potrzeb, to jednak zaprojektowanie całkowicie nowego układu jest pracochłonne³.
- Trudno jest tworzyć dokumenty o nieokreślonej, bałaganiarskiej strukturze.
- Twój chomik może nie być w stanie, mimo kilku obiecujących kroków wstępnych, w pełni pojąć koncepcję znakowania logicznego.

1.3. Plik źródłowy

Plik źródłowy \LaTeX a to zwykły plik tekstowy (plik ASCII). Taki plik można utworzyć w dowolnym edytorze tekstowym. Powinien on zawierać tekst dokumentu oraz instrukcje dla \LaTeX a określające, jak tekst ma zostać złożony.

1.3.1. Odstęp

Znaki *niewidoczne*, takie jak odstępy (spacje) lub znaki tabulacji, są przez \LaTeX a traktowane jednakowo – jako odstęp. *Kolejno* po sobie występujące znaki odstępu \LaTeX traktuje jak *pojedynczy* odstęp. Znaki odstępu znajdujące się na początku wiersza są prawie zawsze ignorowane. Pojedynczy koniec linii jest traktowany jak odstęp.

Pusty wiersz pomiędzy dwoma wierszami tekstu oznacza koniec akapitu. Kolejno występujące puste wiersze \LaTeX traktuje jak jeden. Przykładem może być poniższy tekst. Po prawej stronie (w ramce) przedstawiono wynik składu, a po lewej – zawartość pliku źródłowego.

³Plotki mówią, że jest to jeden z ważniejszych problemów, nad jakim pracują twórcy systemu $\LaTeX 3$.

Rozważmy przykład tabeli zawierającej najważniejsze „parametry” słynnych wzniesień i przełęcz:

```
\begin{tabular}{|l|r|r|r|r|r|}\hline
Nazwa & \multicolumn{2}{|c|}{Wys. w m n.p.m.} & & & \\
Długość & \multicolumn{2}{|c|}{Nachylenie \%} & & & \\
\cline{2-3} \cline{5-6} \% \cline można wstawiać wielokrotnie
& początek & koniec & w km & śr. & max \\ \hline
Col du Galibier & 1401 & 2646 & 18,1 & 6,9\% & 14,5 \\ \hline
Alpe D'Huez & 724 & 1815 & 14,2 & 7,7\% & 15,0 \\ \hline
Passo Gavia & 1734 & 2618 & 13,5 & 6,5\% & 20,0 \\ \hline
\end{tabular}
```

Złożone tabele wyglądają następująco:

Nazwa	Wys. w m n.p.m.		Długość w km	Nachylenie %	
	początek	koniec		śr.	max
Col du Galibier	1401	2646	18,1	6,9%	14,5
Alpe D'Huez	724	1815	14,2	7,7%	15,0
Passo Gavia	1734	2618	13,5	6,5%	20,0

Jeżeli tabela ma być złożona na przykład na szerokość łamu, to zastępujemy `tabular` jego wersją z gwiazdką oraz modyfikujemy preambułę tabeli:

```
\begin{tabular*}{\textwidth}{%
\extracolsep{\stretch{1}}|l|r|r|r|r|}\hline ...
\end{tabular}
```

Złożona tabela wygląda następująco:

Nazwa	Wys. w m n.p.m.		Długość w km	Nachylenie %	
	początek	koniec		śr.	max
Col du Galibier	1401	2646	18,1	6,9%	14,5
Alpe D'Huez	724	1815	14,2	7,7%	15,0
Passo Gavia	1734	2618	13,5	6,5%	20,0

Wewnątrz otoczenia `tabular` \LaTeX odziera poszczególne kolumny stałym odstępem równym `\tabcolsep`. Do składania tabel o określonej szerokości należy ten odstęp zamienić na taki, którego wielkość może się zmieniać (por. punkt 6.3.3). Do tego celu należy użyć wspomnianej w punkcie 2.11.6 (s. 39) instrukcji `@`. Ponadto polecenie `\extracolsep`, umieszczone wewnątrz instrukcji `@{...}`, wstawia *dotatkowy odstęp* między kolejnymi kolumnami – do odwołania poleceniem `\extracolsep` albo aż do końca tabeli.

W powyższym przykładzie polecenie `@{...}` usuwa domyślny odstęp międzykolumnowy, zastępując go specjalnym odstępem o zmiennej wielkości (taki odstęp wstawia polecenie `\stretch`). Manipulując odstępem międzykolumnowym, \LaTeX dopasowuje szerokość tabeli do żądanej wielkości. Otoczenie `tabular*` powinno zawierać w specyfikacji formatu kolumn co najmniej jedną konstrukcję `@{\extracolsep{\stretch{1}}}`. Uważny czytelnik dostrzeże jednak, że kreski poziome pod pierwszym wierszem między

Czytałem, że Knuth dzieli użytkowników systemu `{\TeX}` na `\TeX{}-ników` oraz `\TeX` pertów. `\`
Dzisiaj jest `\today`.

Czytałem, że Knuth dzieli użytkowników systemu `TEX` na `TEX`ników oraz `TEX`pertów. `\`
Dzisiaj jest 21 stycznia 2007.

Niektóre instrukcje `LATEX`owe mają argumenty. Podaje się je w nawiasach klamrowych `{ }`, każdy w osobnej parze nawiasów. Liczba oraz kolejność argumentów jest istotna i wynika z definicji instrukcji. Instrukcje mogą mieć także argumenty opcjonalne, podawane w nawiasach kwadratowych `[]`. W wypadku większej liczby argumentów opcjonalnych oddziela się je przecinkami. Kolejność argumentów opcjonalnych *nie odgrywa roli*.

Poniższe przykłady ilustrują postać instrukcji `LATEX`owych. Ich znaczenie jest tu nieistotne i zostanie opisane później.

Możesz na mnie `\textsl{polegać}`!

Możesz na mnie *polegać*!
Tu wstaw zmianę wiersza. `\newline`
Dziękuję.

Tu wstaw zmianę wiersza. `\newline`
Dziękuję.

1.3.4. Komentarze

Po napotkaniu znaku `%` `LATEX` ignoruje resztę bieżącej wiersza (łącznie ze znakiem końca wiersza) oraz znaki odstępów na początku następnego. Znak `%` jest używany do umieszczania komentarzy i dodatkowych informacji w pliku źródłowym.

Mao zmarł `%` sprawdzić!
`w~1975` roku.

Mao zmarł w 1975 roku.

Znaku `%` używa się niekiedy do dzielenia bardzo długich linii w pliku wejściowym, gdy niedozwolone jest użycie spacji lub złamanie wiersza.

1.4. Struktura pliku źródłowego

`LATEX` oczekuje, że plik źródłowy posiada określoną strukturę. W szczególności, każdy plik źródłowy składa się z dwóch części: preambuły oraz części głównej. Preambuła powinna się rozpoczynać od instrukcji `\documentclass`:

```
\documentclass{...}
```

Instrukcja ta określa rodzaj tworzonego dokumentu. Po niej można umieścić polecenia dotyczące stylu całego dokumentu oraz dołączyć pakiety poszerzające możliwości `LATEX`a. Pakiety dołącza się poleceniem `\usepackage`:

```
\usepackage{...}
```

```
\makebox[.5\width][l]{oooooooo}%  
\makebox{xxxxxxx}
```

ooooooooxxxx

Polecenie `\framebox` działa dokładnie jak `\makebox`, z tym że naokoło pudełka kreślona jest ramka.

Oto przykład zastosowania poleceń `\makebox` i `\framebox`:

```
\makebox[\textwidth]{%  
p o ś r o d k u}\par  
\makebox[\textwidth][s]{%  
r o z s t r z e l o n y}\par  
\framebox[1.1\width]{Teraz jestem Trochę tu jest za szeroko  
obramowany!} \par  
\framebox[0.8\width][r]{Trochę tu jest za szeroko} \par  
\framebox[1cm][l]{Nie ma sprawy}  
Da się to czytać?
```

p o ś r o d k u
r o z s t r z e l o n y
Teraz jestem obramowany!
Trochę tu jest za szeroko
Nie ma sprawy
Da się to czytać?

Pudełka można też przesuwać w pionie. Służy do tego polecenie:

```
\raisebox{przesunięcie}[wysokość][głębokość]{tekst}
```

Argument *przesunięcie* określa wielkość przesunięcia w górę (lub w dół, jeżeli wielkość przesunięcia jest ujemna). Ponadto za pomocą parametrów opcjonalnych *wysokość* oraz *głębokość* można zadać nominalną wysokość oraz głębokość pudełka (`LATEX` będzie traktował pudełko tak, jakby miało zadane wymiary, bez względu na wymiary naturalne). Wewnątrz parametrów można skorzystać z wielkości `\width`, `\height`, `\depth` oraz `\totalheight`.

```
\raisebox{0pt}[0pt][0pt]{\Large%  
\textbf{Aaaa}\raisebox{-0.3ex}{a}%  
\raisebox{-0.7ex}{aa}%  
\raisebox{-1.2ex}{r}%  
\raisebox{-2.2ex}{g}%  
\raisebox{-4.5ex}{h}}  
Krzyczała, ale nikt nie zauważył,  
że coś się jej przytrafiło.
```

Aaaaaaa, Krzyczała, ale nikt nie zauważył, że coś się jej przytrafiło.
h

6.7. Kreski i podpory

W wyniku wykonania polecenia `\rule`, postaci:

```
\rule[przesunięcie]{szerokość}{wysokość}
```

```

\documentclass[a4paper,11pt]{article}
\usepackage{latexsym}
\usepackage[MeX]{polski}
\usepackage[latin2]{inputenc}% ew. utf8 lub cp1250
% Zdefiniowanie autora i~tytułu:
\author{H.~Szczególny}
\title{Minimalizm}
\frenchspacing
\begin{document}
% Wstawienie autora i~tytułu do składu:
\maketitle
% Wstawienie spisu treści:
\tableofcontents
\section{Kilka spostrzeżeń na wstępie}
Właśnie tu zaczyna się mój cudowny artykuł.
\section{Na pożegnanie}
\ldots{} A~tu się on kończy.
\end{document}

```

Rysunek 1.2: Przykład artykułu do czasopisma. Użyte w nim polecenia zostaną objaśnione w dalszej części.

nowe dystrybucje Linuksa), kodowanie CP 1250 (MS Windows) lub ISO 8859-2 (Linux) albo jeszcze coś innego (np. Mac). Dokładna postać wiersza `\usepackage[...]{inputenc}` zależy od wybranego kodowania; wewnątrz nawiasów kwadratowych należy wpisać albo `utf8` jeżeli używamy *Unicode*⁵, albo `latin2` jeżeli korzystamy z ISO 8859-2 lub też `cp1250` jeżeli używamy CP 1250⁶. Więcej informacji na temat kodowania znajdziesz w punkcie 2.5. Nie używaj odstępów w nazwie pliku, przynajmniej na początku swojej przygody z \LaTeX em, bo może ci to skomplikować życie. Wybierając nazwę dla pliku, podaj jako jej rozszerzenie `.tex`. Poprawną nazwą byłoby na przykład `foo.tex`.

2. Uruchom program `latex` na pliku utworzonym w punkcie 1, wpisując w wierszu poleceń⁷:

```
latex foo.tex
```

⁵Uwaga: \LaTeX potrafi przetwarzać dokumenty Unicodowe w ograniczonym zakresie, wystarczającym dla tekstów w językach europejskich, ale dalekim od kompletności.

⁶Wybierz kodowanie ANSI aby zapisać dokument w standardzie CP 1250.

⁷Ponieważ ten sposób pracy jest zwykle nieznanym użytkownikom systemu MS Windows, przypominamy, że wiersz poleceń możemy uruchomić na przykład tak: wywołujemy *Start/Uruchom*, wpisujemy `cmd` lub `command` i naciskamy klawisz Enter.

6.5. Więcej o odległościach

Kiedy to tylko możliwe, unikajmy stosowania wymiarów zdefiniowanych w jednostkach absolutnych, takich jak punkty czy milimetry. Starajmy się raczej odnosić wymiary do już istniejących, takich jak wysokość czy szerokość kolumny. W poniższym przykładzie szerokość rysunku jest definiowana jako połowa szerokości bieżącej szpalty:

```
\includegraphics[width=0.5\textwidth]{sowauszata.eps}
```

Następujące trzy polecenia pozwalają określić szerokość, wysokość i głębokość *tekstu*:

```

\settoheight{nazwa}{tekst}
\settodepth{nazwa}{tekst}
\settowidth{nazwa}{tekst}

```

Oto przykład ilustrujący zastosowanie tych poleceń:

```

\newenvironment{vardesc}[1]{%
\settoheight{\parindent}{#1:\ }
\makebox[0pt][r]{#1:\ }}{}

```

```

\begin{displaymath} a^2+b^2=c^2
\end{displaymath}

```

```

\begin{vardesc}{gdzie}%
$a$, $b$ -- przyprostokątne, \par
$c$ -- przeciwprostokątna.
\end{vardesc}

```

$$a^2 + b^2 = c^2$$

gdzie: a , b – przyprostokątne,
 c – przeciwprostokątna.

6.6. Pudełka

Każdą stronę \LaTeX tworzy z pudełek, które odpowiednio skleja. Elementarnymi pudełkami są litery, z których skleja się słowa. Słowa są następnie łączone w wiersze, a wiersze – w akapity. Do łączenia używany jest specjalny klej, który dzięki elastyczności pozwala wyrazić ścisnąć lub rozciągnąć tak, by dokładnie wypełniały wiersze na stronie.

Trzeba przyznać, że takie ujęcie jest mocno uproszczoną wersją tego, co się naprawdę dzieje, chociaż zasadniczo biorąc, działanie \TeX a można jednak wyjaśnić właśnie w terminach pudełek oraz kleju (odstępu wstawianego między pudełkami). Pudełkami są nie tylko litery. Do pudełka można włożyć praktycznie wszystko, także inne pudełka. Każde pudełko \LaTeX traktuje jak pojedynczą literę.

Chociaż nie mówiliśmy o tym wprost, pudełka pojawiały się już w poprzednich rozdziałach. Na przykład polecenie `\includegraphics` albo otoczenie `tabular` tworzą pudełka. Dzięki temu dwa rysunki albo tabele można

`report` dłuższe opracowania, dysertacje magisterskie i doktorskie...
`book` książki.
`letter` listy.

Opcje pozwalają zmieniać sposób działania klas. Poszczególne opcje rozdziela się przecinkami. Najczęściej wykorzystywane opcje dla klas standardowych to:

`10pt`, `11pt`, `12pt` Ustalenie stopnia pisma dla tekstu zasadniczego dokumentu. Domyślną wartością jest 10 punktów.

`a4paper`, `letterpaper`, ... Ustalenie wymiarów papieru. Wartością domyślną jest `letterpaper`. Inne dopuszczalne wartości to: `a5paper`, `b5paper`, `executivepaper` i `legalpaper`.

`fleqn` Składanie wyeksponowanych wzorów matematycznych od lewego marginesu zamiast domyślnego centrowania.

`leqno` Umieszczanie numerów wzorów matematycznych na lewym marginesie zamiast domyślnie na prawym.

`titlepage`, `notitlepage` Pierwsza z opcji powoduje, że \LaTeX składa tytuł (instrukcja `\maketitle`) oraz streszczenie (instrukcja `\abstract`) na oddzielnej stronie, druga – że skład tekstu zaczyna się na stronie tytułowej. W klasie `article` tytuł i streszczenie nie są domyślnie składane na oddzielnych stronach, podczas gdy w stylach `report` i `book` – są.

`onecolumn`, `twocolumn` Skład jedno- lub dwukolumnowy (dwukolumnowy).
`oneside`, `twoside` Druk na jednej lub na dwóch stronach kartki papieru.

W klasach `article` i `report` domyślną opcją jest `oneside`, natomiast w klasie `book` – `twoside`. Włączenie opcji `oneside` powoduje przy okazji, że \LaTeX nie wyrównuje wysokości kolejnych stron, dopuszczając pewną ich zmienność.

`openright`, `openany` Wybranie pierwszej opcji powoduje, że tytuły rozdziałów będą umieszczane na stronach nieparzystych. W klasie `article` opcja nie ma znaczenia, gdyż w tej klasie nie jest zdefiniowane pojęcie rozdziału. W klasie `report` domyślną wartością jest `openany`, a w klasie `book` – `openright`.

Przykład. Plik źródłowy może się rozpoczynać od następującej instrukcji:

```
\documentclass[11pt,twoside,a4paper]{article}
```

W tym wypadku dokument zostanie złożony w klasie `article`, pismem w stopniu *11 punktów*, i zostanie przygotowany do wydruku po *dwóch* stronach kartki papieru formatu *A4*.

1.6.2. Pakiety

Pakiety rozszerzają możliwości \LaTeX a. Sam \LaTeX nie ma na przykład instrukcji do dołączania grafiki, kolorowania tekstu, łamania dużych tabel

6.4. Układ graficzny strony

Wymiary papieru można podać jako argumenty instrukcji `\documentclass`. Na podstawie zadeklarowanych wymiarów \LaTeX oblicza szerokość i wysokość kolumny, marginesy i inne parametry. Na rysunku 6.2 przedstawiono dostępne parametry graficznego układu strony. Do przygotowania rysunku użyliśmy pakietu `layout` z zestawu „tools”⁸. Jeżeli obliczone przez \LaTeX a wartości są z pewnych względów nieodpowiednie, to można je zmienić.

Zanim jednak začniemy eksperymentować, zwiększając na przykład szerokość szpalty, chwilę pomyślmy. Jak dla większości rzeczy w \LaTeX u, istnieją ważne powody, dlaczego szerokość szpalty jest taka a nie inna.

Z pewnością w porównaniu z wydrukiem przygotowanym za pomocą świeżo kupionego programu MS Word strona \LaTeX owa ma denerwująco wąską szpaltę. Ale spójrzmy na książkę z renomowanego wydawnictwa i policzmy na niej liczbę znaków w przeciętnym wierszu. Okazuje się, że wynosi ona około 66. Jeśli porównamy ją z wydrukiem złożonym przez \LaTeX a, to zapewne i tym razem będzie ona zbliżona do 66. Z doświadczeń wynika bowiem, że w miarę wzrostu liczby znaków w wierszu czytanie staje się męczące. Dzieje się tak, gdyż przy długich wierszach naszym oczom trudniej jest przenosić wzrok z końca jednego wiersza na początek następnego. Jest to jedna z przyczyn stosowania składu wielołamowego w gazetach i czasopiśmie.

Tak więc, jeśli zwiększamy szerokość kolumny, to pamiętajmy, że może to utrudnić odbiorcom czytanie naszej pracy. No, ale dość już kazań. Obiecaliśmy przecież wyjaśnić, jak można te rzeczy robić...

W \LaTeX u mamy dwie instrukcje do zmiany wielkości wymiarów, używane zazwyczaj w obrębie preambuły dokumentu. Pierwsza z nich *nadaje* parametrowi określoną wielkość:

```
\setlength{parametr}{wielkość}
```

Drugie polecenie *zwiększa* wartość parametru o określoną wielkość:

```
\addtolength{parametr}{wielkość}
```

Z tej pary częściej stosowana jest druga instrukcja, ponieważ pozwala zmieniać wymiary. Przykładowo, aby zwiększyć szerokość szpalty o jeden centymetr, umieszczamy w preambule dokumentu następujące polecenia:

```
\addtolength{\hoffset}{-0.5cm}
\addtolength{\textwidth}{1cm}
```

Zmianę parametrów układu graficznego strony ułatwia pakiet `geometry`. W wykonywaniu operacji arytmetycznych na wymiarach pomaga pakiet `calc`.

⁸CTAN://macros/latex/packages/tools.

Tabela 1.1: Wybrane pakiety z podstawowej dystrybucji L^AT_EXa

doc	Służy do drukowania dokumentacji pakietów oraz innych części składowych L ^A T _E Xa. Opis znajduje się w pliku <code>doc.dtx</code> ^a .
exscale	Umożliwia skalowanie fontów matematycznych, tak by optycznie były zgodne z otaczającym tekstem, np. w tytułach rozdziałów. Opis w <code>ltxscale.dtx</code> .
fontenc	Definiuje układ znaków, którego ma używać L ^A T _E X. Opis w punkcie 2.5 i w <code>loutenc.dtx</code> .
ifthen	Umożliwia korzystanie z poleceń typu <code>if... then do... otherwise do</code> . Opis w <code>ifthen.dtx</code> i [6].
latexsym	Udostępnia specjalny font symboliczny (fonty <code>lasy</code>). Opis w <code>latexsym.dtx</code> i [6].
makeidx	Udostępnia polecenia do przygotowywania skorowidzów. Opis w punkcie 4.3 i [6].
syntonly	Powoduje, że dokument jest przetwarzany bez składania czegokolwiek. Przydatny do szybkiego sprawdzenia, czy dokument nie zawiera błędów. Opis w <code>syntonly.dtx</code> i [6].
inputenc	Definiuje układ znaków w pliku źródłowym, jak: ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM, Apple Macintosh, Next, ANSI-Windows, itd. Opis w <code>inputenc.dtx</code> .

^aPlik ten powinien być zainstalowany w twoim systemie. Aby otrzymać z niego plik DVI, wystarczy w katalogu z prawem do zapisu napisać `latex doc.dtx`. To samo stosuje się do innych pakietów z tej tabeli.

Punkt 4.4 na stronie 71 niniejszego *Wprowadzenia* oraz podręcznik [6] w punkcie 4.4 na stronie 54 zawierają więcej informacji o paginach, w szczególności o sposobach samodzielnego definiowania ich wyglądu.

1.7. Nazwy plików związanych z L^AT_EXem

Pracując z L^AT_EXem, szybko zauważysz, że na dysku pojawia się mnóstwo plików o różnych rozszerzeniach nazwy, a ty nie wiesz, co to za jedno. W poniższym wykazie objaśniono rozmaite typy plików, z którymi możesz się zetknąć. Wykaz ten nie pretenduje do kompletnego, dlatego napisz do nas, gdy napotkasz jakieś nowe rozszerzenie, które uznasz za warte opisanie.

- .**tex** Plik źródłowy z dokumentem w notacji L^AT_EXa bądź zwykłego T_EXa. Można go kompilować programem `latex` bądź, odpowiednio, `tex`.
- .**sty** Pakiet makr L^AT_EXowych. Plik tego typu można dołączać do dokumentu L^AT_EXowego, używając do tego celu instrukcji `\usepackage`.

Jeżeli taki odstęp, w wyniku złamania akapitu na wiersze, wypadnie na początku lub na końcu wiersza, to zostanie on usunięty – aby zapobiec justowaniu akapitu „w chorągiewkę”. Jeżeli L^AT_EX ma wstawić odstęp także na początku lub na końcu wiersza, to zamiast `\hspace` należy użyć „gwiazdkowej” wersji `\hspace*`. Argument *odległość* oznacza wymiar L^AT_EXowy. W najprostszej postaci jest to liczba wraz z jednostką odległości. Wykaz ważniejszych spośród dostępnych w L^AT_EXu jednostek odległości znajduje się w tabeli 6.5.

To jest `\hspace{1.5cm}`odstęp
równy 1,5 cm.

To jest odstęp równy 1,5 cm.

Tabela 6.5: L^AT_EXowe jednostki miary

mm	milimetr $\approx 1/25$ cala	▯
cm	centymetr = 10 mm	▯
in	cal = 25,4 mm	▯
pt	punkt $\approx 1/72$ cala $\approx \frac{1}{3}$ mm	▯
em	w przybliżeniu szerokość „M” w bieżącym foncie	▯
ex	w przybliżeniu wysokość „x” w bieżącym foncie	▯

Często wygodnie jest użyć odległości „elastycznej”, zostawiając L^AT_EXowi nieco swobody doboru takiej odległości, jaką uzna za najlepszą z punktu widzenia jakości składu. Taką elastyczną odległość zapisujemy następująco: *n plus p minus m*. Części „plus *p*” i „minus *m*” są opcjonalne (każdą z nich można pominąć). Tego typu odległości mają naturalną wielkość *n* i mogą się kurczyć lub rozciągać w zakresie od *n – m* do *n + p*.

Omawiany wcześniej odstęp między akapitami (`\parskip`) jest przykładem L^AT_EXowego wymiaru o zmiennej wielkości. Część wymiarów może mieć wartości zmienne, część jednak musi mieć wartość stałą. Powinno być na przykład zrozumiałe, że wcięcie akapitowe musi być wielkością stałą, podobnie jak szerokość i wysokość lamu.

Polecenie:

`\stretch{n}`

wstawia specjalny rozciągliwy odstęp, który potrafi wypełnić całą wolną przestrzeń w pionie lub w poziomie. Jeżeli na przykład wstawimy w wierszu dwa lub więcej poleceń `\hspace{\stretch{n}}`, to odstępy dzięki nim uzyskane będą miały wielkość według proporcji zadanych przez argument *n*. W poniższym przykładzie odstęp między *x* a *y* jest trzy razy mniejszy od odstępu między *y* a *z*.

1.8. Duże dokumenty

Pracując nad dużym dokumentem, wygodnie jest podzielić plik źródłowy na mniejsze części. W \LaTeX u mamy dwie instrukcje ułatwiające pracę z tak podzielonymi dokumentami. Pierwszą z nich jest:

```
\include{plik}
```

Włącza ona do dokumentu zawartość innego pliku. Przed przetworzeniem, a także po przetworzeniu zawartości włączanego pliku \LaTeX rozpoczyna nową stronę.

Drugiej instrukcji używa się w preambule. Pozwala ona wstawiać do tekstu jedynie wybrane pliki.

```
\includeonly{plik1,plik2,...}
```

Spośród instrukcji `\include` zostaną wykonane tylko te, które dotyczą plików wymienionych w argumencie `\includeonly`. Uwaga: w wykazie plików nie wolno używać odstępów; poszczególne pliki należy oddzielać wyłącznie przecinkiem!

Polecenie `\include` rozpoczyna skład dołączanego tekstu od nowej strony. W połączeniu z poleceniem `\includeonly` w preambule instrukcja `\include` umożliwia przetwarzanie wybranych plików bez zmiany miejsc łamania poszczególnych stron i z zachowaniem prawidłowej numeracji stron, rozdziałów, tabel itp. Czasami jednak rozpoczynanie składu od nowej strony nie jest pożądane. W takiej sytuacji zamiast `\include` należy posłużyć się instrukcją:

```
\input{plik}
```

Wstawia ona zawartość podanego pliku już bez żadnych dodatkowych efektów.

Argument poleceń `\input` i `\include` może zawierać pełną ścieżkę do pliku, ale w imię wygody i przenośności nie należy używać ścieżek bezwzględnych. Na przykład:

```
\include{C:\Documents and Settings\elka\Moje dokumenty\r2.tex}
```

jest koszmarnym błędem w większości dystrybucji \LaTeX a. Nawet jeżeli nasz \LaTeX wie, co oznacza specyficzny dla MS Windows zapis `C:\`, oraz poradzi sobie z odstępami w nazwach katalogów i ze znakami `\` (które w tym wypadku nie są początkiem żadnego polecenia), to taki dokument przesłany komuś, kto będzie go kompilował w systemie Linux, sprawi mu mnóstwo problemów. Pamiętajmy: bez względu na to, jakiego systemu operacyjnego używamy, katalogi powinniśmy oddzielać znakiem `/`.

do typowego dokumentu składanego na przykład edytorem MS Word; drugi z kolei będzie się doskonale nadawał do dokumentu PDF, który chcemy umieścić w sieci WWW, z uwagi na to, że krój Palatino – reprodukowany w niskiej rozdzielczości – jest dużo bardziej czytelny niż CM, a nawet *Times New Roman*. Uwaga: aby powyższe deklaracje zadziałały, instalacja \LaTeX a musi zawierać ww. fonty³.

6.3. Odstępy

6.3.1. Zmiana wielkości interlinii

Wielkość odstępów między wierszami dokumentu można zmienić, umieszczając w preambule polecenie `\linespread`, postaci:

```
\linespread{czynnik}
```

Parametr *czynnik* określa powiększenie odstępu między wierszami. Znany z maszyn do pisania efekt podwójnej interlinii, czyli podwojenie odstępu, uzyskamy za pomocą `\linespread{1.6}`. Aby otrzymać odstęp wielkości 1,5, powinniśmy wpisać `\linespread{1.3}`. Pojedynczemu i zarazem domyślnemu odstępowi odpowiada wartość 1.

Polecenie `\linespread` wpływa na odstępy międzywierszowe w całym dokumencie. Jeśli są wyraźne powody do zmiany odstępu tylko w pewnym fragmencie dokumentu, to lepsza może się okazać instrukcja:

```
\setlength{\baselineskip}{1.5\baselineskip}
```

Poniższy przykład ilustruje wykorzystanie `\baselineskip`:

```
{\setlength{\baselineskip}%
  {1.5\baselineskip}
Ten akapit jest składany
z~\ci{\baselineskip} ustawionym
na 1,5 dotychczasowej wartości.
Zwróćmy uwagę na wystąpienie
\ci{\par} na końcu akapitu.\par}
```

Przeznaczenie tego akapitu jest jasne. Ilustruje on, że po zamykającym nawiasie klamrowym następuje powrót do normalnego składu.

Ten akapit jest składany z `\baselineskip` ustawionym na 1,5 dotychczasowej wartości. Zwróćmy uwagę na wystąpienie `\par` na końcu akapitu. Przeznaczenie tego akapitu jest jasne. Ilustruje on, że po zamykającym nawiasie klamrowym następuje powrót do normalnego składu.

³Wchodzą one w skład współczesnych dystrybucji \LaTeX a. Jeżeli ich nie masz, to powinieneś je doinstalować. W dokumentacji dystrybucji powinno być opisane, jak się to robi.

Zobaczmy, dlaczego w poniższych przykładach w jednej z takich sytuacji należy przejść do nowego akapitu, a w innej – nie. Czytelnik, który nie zna jeszcze wszystkich poleceń użytych w tych przykładach, powinien dokładnie przeczytać ten i następny rozdział, a następnie wrócić do tego punktu i przeczytać go jeszcze raz.

```
% Przykład 1
\ldots Słynne równanie Einsteina
\begin{equation}
e = m \cdot c^2 \ ; \ ;
\end{equation}
jest najbardziej znanym, ale też
najmniej rozumianym równaniem w fizyce.
```

```
% Przykład 2
\ldots którego wynikiem jest prądowe prawo Kirchhoffa:
\begin{equation}
\sum_{k=1}^n I_k = 0 \ ; \ .
\end{equation}
```

Napięciowe prawo Kirchhoffa ma zaś swój początek w \ldots

```
% Przykład 3
\ldots co ma określone zalety.
```

```
\begin{equation}
I_D = I_F - I_R
\end{equation}
jest rdzeniem innego modelu tranzystora. \ldots
```

Mniejszą od akapitu jednostką podziału tekstu jest zdanie. W tekstach angielskojęzycznych odstęp po kropce kończącej zdanie jest większy od odstępu po kropce oznaczającej skrót. Zależnie od kontekstu \LaTeX stara się użyć krótszego lub dłuższego odstępu. W razie pomyłek z jego strony powinniśmy mu wskazać nasze intencje. Jak to zrobić, wyjaśniamy w dalszej części tego rozdziału.

Właściwa organizacja tekstu dotyczy nawet fragmentów zdań. Wiele języków ma bardzo skomplikowaną interpunkcję, ale w większości wypadków (wliczając angielski i niemiecki) stawiając przecinek w określonym miejscu w zdaniu, nie popełnimy na ogół błędu, pamiętając o zasadzie, że przecinek oznacza krótką przerwę w wypowiedzi. Dlatego – jeśli nie jesteśmy pewni, gdzie w zdaniu należy go postawić – przeczytajmy zdanie na głos i postavmy przecinki wszędzie tam, gdzie zrobiliśmy krótką przerwę na wdech. Ale nie sugerujemy się wyłącznie tym! Jeśli w danym miejscu przecinek nie wygląda najlepiej, to go usuńmy; jeśli w innym miejscu odczuwamy potrzebę

Tabela 6.4: Polecenia wyboru fontów w trybie matematycznym

Polecenie	Przykład	Wynik
<code>\mathcal{...}</code>	<code>\$\$\mathcal{B}=c\$</code>	$\mathcal{B} = c$
<code>\mathrm{...}</code>	<code>\$\$\mathrm{K}_2\$</code>	K_2
<code>\mathbf{...}</code>	<code>\$\$\sum x=\mathbf{v}\$</code>	$\sum x = \mathbf{v}$
<code>\mathsf{...}</code>	<code>\$\$\mathsf{G\times R}\$</code>	$G \times R$
<code>\mathtt{...}</code>	<code>\$\$\mathtt{L}(b,c)\$</code>	$L(b, c)$
<code>\mathnormal{...}</code>	<code>\$\$\mathnormal{R_{19}}\neq R_{19}\$</code>	$R_{19} \neq R_{19}$
<code>\mathit{...}</code>	<code>\$\$\mathit{ffi}\neq ffi\$</code>	$\mathit{ffi} \neq ffi$

6.2.2. Uwaga, niebezpieczeństwo!

Jak zaznaczyliśmy na początku rozdziału, nie należy instrukcji zmiany fontu wstawiać *explicite* do pliku źródłowego. Byłoby to niezgodne z podstawową ideą \LaTeX a, jaką jest oddzielenie formy od treści dokumentu i posługiwanie się formatowaniem logicznym, a nie wizualnym. Jeżeli fragment tekstu ma zostać wyróżniony przez złożenie go innym krojem lub stopniem pisma, to należy zdefiniować odpowiednie polecenie i potem właśnie jego używać w treści dokumentu.

```
% w~preamble albo pakiecie
\newcommand{\uwaga}[1]{\textbf{\#1}}
% po \begin{document}
\uwaga{Bacność!} Przewody sieci
trakcyjnej są pod napięciem.
Dotknięcie grozi \uwaga{śmiercią}.
```

Bacność! Przewody sieci trakcyjnej są pod napięciem. Dotknięcie grozi **śmiercią**.

Niewątpliwą zaletą tego podejścia jest to, że kiedy później będziemy chcieli wyróżnić wszystkie elementy, na które czytelnik powinien zwrócić *szczególną uwagę*, w sposób inny niż składając je pismem półgrubym, to nie musimy przeglądać całego pliku w celu sprawdzenia, czy dane wystąpienie `\textbf` dotyczy tekstu, na który ma zostać zwrócona *szczególna uwaga*, czy też wstawione zostało w zupełnie innym celu.

Na zakończenie rada z gatunku estetycznych: nie należy przesadzać ze stosowaniem wielu różnych krojów pisma w jednym dokumencie.

6.2.3. Użycie alternatywnych krojów pisma

Większość dokumentów jest składanych w \LaTeX u z użyciem domyślnego kroju, będącego repliką *Computer Modern*. Jeżeli wygląd znaków z rodziny CM nam się znuży, to możemy złożyć dokument innym krojem. Musimy jednak pamiętać, że na ogół alternatywne kroje nie są tak kompletne jak

oznaczają, odpowiednio: zachętę do złamania wiersza, niezgodę na złamanie wiersza, zachętę do złamania strony i niezgodę na złamanie strony (w miejscu ich wystąpienia w dokumencie). Opcjonalny argument n , o dopuszczalnej wartości od 0 do 4, określa stopień tej zachęty (niezгоды). Domyślna wartość 4 to bezwarunkowy zakaz lub nakaz złamania linii/strony. Wartość mniejsza od 4 pozostawia L^AT_EX-owi swobodę zignorowania instrukcji, jeżeli skład otrzymany w jej rezultacie byłby kiepskiej jakości.

Polecen z grupy `break` nie należy mylić z tymi z grupy `new`. Mimo otrzymania polecenia typu `break` L^AT_EX stara się wyrównać wiersz do prawego marginesu czy też wypełnić stronę do całej jej wysokości. Nietrudno zgadnąć, jakiej instrukcji należy użyć, gdy naprawdę zależy nam na rozpoczęciu nowego wiersza¹.

Jeżeli w wierszu zakończonym instrukcją `\newline` jest zbyt mało tekstu, to L^AT_EX nie wyrówna tego wiersza do prawego marginesu, lecz wstawi po tekście odpowiedni odstęp. Jeżeli zamiast `\newline` użyjemy `\linebreak`, to L^AT_EX postara się wyrównać zawartość kończącej linijki do prawego marginesu. Zbyt małe wypełnienie wiersza tekstem zmusi L^AT_EXa do umieszczenia między wyrazami zbyt dużych odstępów. Efekt ten L^AT_EX sygnalizuje podczas przetwarzania dokumentu, wyświetlając komunikat:

```
Underfull \hbox (badness 10000) in paragraph at lines 4--5
```

Liczba po słowie `badness` w tym komunikacie wskazuje, jak bardzo L^AT_EXowi „nie podoba się” złożony wiersz. Tutaj jest to maksymalna negatywna ocena 10000. Liczby na końcu komunikatu to numery pierwszej i ostatniej linii akapitu, w którym L^AT_EX musiał złożyć wiersz z nadmiernymi odstępami między wyrazami. Chociaż wielu użytkowników L^AT_EXa nie zwraca uwagi na podobne ostrzeżenia, warto sobie zdawać sprawę, co one oznaczają.

Oprócz wyżej wymienionych istnieją jeszcze w L^AT_EXu instrukcje:

```
\clearpage, \cleardoublepage
```

Obydwie rozpoczynają skład nowej strony. Instrukcja `\cleardoublepage` działa tak jak `\clearpage`, z tym że rozpoczynana strona ma numer nieparzysty; w razie potrzeby tworzona jest strona pusta (wakatowa)². W trybie składu dwukolumnowego (opcja `twocolumn`) instrukcja `\newpage` kończy łam, natomiast `\clearpage` oraz `\cleardoublepage` kończą stronę, pozostawiając w razie potrzeby pusty prawy łam.

Jeżeli na stronie zakończonej instrukcją `\newpage` albo `\clearpage` jest zbyt mało tekstu, to L^AT_EX wstawia odpowiedni odstęp u dołu strony, wypełniający pozostałą część kolumny. W wypadku polecenia `\pagebreak` L^AT_EX wyrównuje zawartość kolumny do dolnego brzegu, wstawiając odstępy

¹Zagadka ta jest łatwa dla znających język angielski – jest to polecenie `\newline`.

²Mówiąc precyzyjnie zależy to od stanu opcji `openright/openany`, w szczególności w klasie „`article`” `\cleardoublepage` domyślnie działa tak jak `\clearpage`.

i interlinię. Każdy z nich można dobrać niezależnie od ustawienia pozostałych. Oznacza to na przykład, że zmiana stopnia pisma nie powoduje zmiany jego kroju ani odmiany.

Tabela 6.1: Polecenia wyboru krojów i odmian

<code>\textrm{...}</code>	krój szeryfowy	<code>\textsf{...}</code>	krój bezszeryfowy
<code>\texttt{...}</code>	grotesk, tj. pismo o jednakowej szerokości znaków		
<code>\textmd{...}</code>	pismo jasne	<code>\textbf{...}</code>	pismo grube
<code>\textup{...}</code>	odmiana prosta	<code>\textit{...}</code>	<i>kursywa</i>
<code>\textsl{...}</code>	odmiana pochyla	<code>\textsc{...}</code>	KAPITALIKI
<code>\emph{...}</code>	<i>wyróżnienie</i>		
<code>\textnormal{...}</code>	główny font dokumentu		

Tabela 6.2: Polecenia jednoczesnego wyboru stopnia pisma i interlinii

<code>\tiny</code>	mikroskopijny	<code>\Large</code>	większy
<code>\scriptsize</code>	bardzo mały	<code>\LARGE</code>	bardzo duży
<code>\footnotesize</code>	mniejszy	<code>\huge</code>	ogromny
<code>\small</code>	mały	<code>\Huge</code>	największy
<code>\normalsize</code>	normalny		
<code>\large</code>	duży		

Przy okazji omawiania poleceń dotyczących fontów trzeba wspomnieć o koncepcji *grupowania*. Grupa zaczyna się od znaku `{`, a kończy znakiem `}`. Grupy służą do ograniczania zasięgu działania poleceń L^AT_EXa. Przyjrzyjmy się następującemu przykładowi:

```
Lubię {\LARGE duże oraz
\small małe} litery} i~cyfry.
```

```
Lubię duże oraz małe litery i cyfry.
```

Pierwszy nawias klamrowy rozpoczyna grupę, potem polecenie `\LARGE` zmienia stopień pisma na *bardzo duży*, w którym zostanie złożony napis „duże oraz”. Kolejny otwierający nawias klamrowy zaczyna następną grupę. W jej obrębie polecenie `\small` zmienia stopień pisma na mały. Do złożenia w tym stopniu przewidziano jedynie słowo „małe”, bo nawias `}` za tym słowem zamyka grupę. Po zamknięciu grupy następuje powrót do stopnia pisma aktualnego przed jej rozpoczęciem, czyli `\LARGE`. W nim zostanie złożone słowo „litery”. Zamknięcie tej grupy powoduje, że resztę tekstu L^AT_EX

a wyrazy-argumenty mogą zawierać (oprócz znaku -) wyłącznie litery. Nie ma natomiast znaczenia, czy w tych wyrazach używa się liter wielkich czy małych. Instrukcja `\hyphenation` z przykładu poniżej pozwala podzielić słowo „czwierć-li-trówka” jedynie w dwóch zaznaczonych miejscach i w ogóle zabrania dzielić słowa „szczypce”. Wyrazy z listy argumentów nie mogą zawierać żadnych znaków specjalnych ani symboli. Przykład:

```
\hyphenation{czwierć-li-trówka szczypce}
```

W językach fleksyjnych, do jakich należy polski, instrukcja `\hyphenation` jest dużo mniej przydatna niż w angielskim. Aby dany wyraz zawsze był dobrze przenoszony, należałoby wypisać wszystkie jego formy. Opracowane przez Hannę Kołodziejską, Bogusława L. Jackowskiego i Marka Ryćko wzorce przenoszenia wyrazów dla języka polskiego sprawdzają się w tak znacznej większości wypadków, że praktycznie nie ma potrzeby stosowania tej konkretnej instrukcji. Co więcej, próba użycia polecenia `\hyphenation` w wypadku stosowania również pakietu `inputenc` (por. punkt 2.5) zakończy się błędem w czasie przetwarzania dokumentu.

Instrukcja `\-` wskazuje, w których miejscach *wolno* \LaTeX owi przenieść wyraz do nowego wiersza; \LaTeX nie podzieli tego wyrazu w żadnym oprócz wskazanych miejsc. Instrukcja odnosi się do konkretnego wystąpienia słowa w dokumencie i nie wpływa na miejsca podziału w innych jego wystąpieniach. Przydaje się ona szczególnie w wypadku wyrazów ze znakami specjalnymi, na przykład akcentowanymi, gdyż automatycznie \LaTeX dzieli jedynie wyrazy złożone z samych liter.

```
Nie\~bie\~sko\~bia\~ło\~zie\~ło\~%
no\nie\~bie\~ski
```

```
Niebieskobiałozielononiebieski
```

Tekst będący argumentem polecenia:

```
\mbox{tekst}
```

nigdy nie zostanie przeniesiony.

```
Numer mojego telefonu wkrótce
się zmieni na \mbox{0116 291 2319}.
```

```
Numer mojego telefonu wkrótce się zmieni na
0116 291 2319.
```

```
Parametr \mbox{\emph{nazwa}} to
nazwa pliku.
```

```
Parametr nazwa to nazwa pliku.
```

Polecenie `\fbox` jest podobne do `\mbox`, z tym że dodatkowo dookoła argumentu rysuje ramkę (por. punkt 6.6).

2.3. Kilka gotowych oznaczeń napisów

W przykładach na poprzednich stronach pojawiło się kilka prostych instrukcji \LaTeX a do składania krótkich napisów.

6.1.3. Nadmiarowe odstępy

Gdy stworzymy nowe otoczenie, problemem mogą być zbędne odstępy, które \LaTeX wstawia do składu. Rozważmy przykład otoczenia, które ma się rozpoczynać od akapitu bez wcięcia, ponadto pierwszy akapit po otoczeniu także nie ma mieć wcięcia. Polecenie `\ignorespaces`, umieszczone jako ostatnie polecenie bloku *begin* otoczenia, spowoduje zignorowanie wszystkich odstępow występujących przed pierwszym akapitem otoczenia. Usunięcie drugiego wcięcia przez umieszczenie w bloku końcowym `\ignorespaces` jest niemożliwe, gdyż zawsze ostatnim poleceniem będzie `\end{otoczenie}`, które anuluje działanie `\ignorespaces`. W takiej sytuacji trzeba skorzystać z polecenia `\ignorespacesafterend`. Napotkawszy je, \LaTeX wstawi `\ignorespaces` dopiero po wykonaniu zamykającego `\end{otoczenie}`.

```
\newenvironment{proste}%
{\noindent}%
{\par\noindent}

\begin{proste}
Zobacz odstępy\z~lewej strony.
\end{proste}
Tak samo\tutaj.
```

```
Zobacz odstępy
z lewej strony.
Tak samo
tutaj.
```

```
\newenvironment{poprawne}%
{\noindent\ignorespaces}%
{\par\noindent%
\ignorespacesafterend}

\begin{poprawne}
Bez odstępu\z lewej strony.
\end{poprawne}
Tak samo\tutaj.
```

```
Bez odstępu
z lewej strony.
Tak samo
tutaj.
```

6.1.4. Własne pakiety

W wypadku definiowania wielu nowych poleceń i otoczeń preambula dokumentu może się znacznie wydłużyć. Dobrze w takiej sytuacji stworzyć pakiet zawierający definicje tych instrukcji i otoczeń. Taki pakiet można później dołączyć do dokumentu poleceniem `\usepackage`.

Tworzenie pakietu polega na skopiowaniu poleceń z preambuły do oddzielnego pliku o rozszerzeniu `.sty`. Na początku pakietu należy wpisać polecenie:

```
\ProvidesPackage{nazwa}
```

Dzięki instrukcji `\ProvidesPackage` \LaTeX poznaje nazwę pakietu, a to pozwala mu na przykład ostrzec użytkownika w wypadku powtórnego do-

nie-	niebie-	niebiesko-	niebiesko-czar-
biesko-czarny	sko-czarny	-czarny	ny

Standardowy L^AT_EX nie zna polskich norm i dlatego dzieli wyrazy wieloczłonowe w miejscu połączenia, bez powielania łącznika. Jeżeli do składu w języku polskim korzystamy z zestawu `platex`, to w pliku źródłowym w miejscu łącznika w wyrazach wieloczłonowych powinniśmy zastosować instrukcję `\dywiz` (np. `niebiesko{\dywiz}czarny`).

W pewnych sytuacjach lepiej nie dzielić wyrazów połączonych łącznikiem. Jeżeli na przykład mówimy o wydziale K-2, kodzie pocztowym czy numerze telefonu, to w takich wypadkach łącznik zapisujemy w pliku źródłowym jako pojedynczą kreskę -.

Znaku łącznika używa się również do przenoszenia wyrazów. Jednak w L^AT_EXu odbywa się to automatycznie i nie wymaga ręcznych ingerencji użytkownika.

Półpauza to kreska o połowę krótsza od myślnika. Stosuje się ją przede wszystkim w zapisie zakresów liczbowych, np. „str. 11–13”, czy „w latach 1960–1963”. Przed i po półpauzie nie dodaje się odstępów. Odstępy takie muszą się pojawić w sytuacjach takich jak: „11 października – 13 listopada”. Półpauzę zapisuje się za pomocą dwóch następujących po sobie minusów --.

Myślnik „—” to dłuższa kreska, używana jako znak przestankowy. Zapisujemy go za pomocą trzech następujących po sobie znaków -, czyli ---. W języku polskim należy przed i po myślniku umieścić odstęp, inaczej niż w krajach anglosaskich, gdzie nie otacza się myślnika odstępami. Wiele osób uważa konstrukcję złożoną z odstępów, myślnika i kolejnego odstępów za zbyt wybijającą się w składzie. Z tychże estetycznych powodów często w roli myślnika używa się „dwukreskowej” półpauzy, i takie właśnie podejście zastosowano w niniejszym tłumaczeniu.

We wzorach matematycznych, czyli wewnątrz trybu matematycznego, znak minusa uzyskujemy, pisząc zwyczajnie -. Przykładowo, zapis `-2` daje w składzie `-2`, podczas gdy `-2` daje `-2`.

2.4.3. Odstępy niełaflliwe

Polskie zasady typograficzne nie pozwalają łamać akapitów z pozostawianiem na końcu wierszy jednoliterowych spółników bądź przyimków. Przykładowo, w zdaniu „Jan Kochanowski urodził się w Czarnolesie” nieładnie na końcu wiersza wyglądałby przyimek „w”.

Odstępy, na których nie wolno złamać wiersza, zaznacza się w pliku źródłowym znakiem tyldy „~” zamiast zwykłym odstępem. Na przykład, aby w powyższej sytuacji zapobiec przeniesieniu składu do nowego wiersza, powinniśmy zapisać w pliku źródłowym: `w~Czarnolesie`.

Jest wiele sytuacji, w których związek fragmentów zdania jest tak silny, że wewnątrz nich nie należy łamać na wiersze. Nie zawsze decyzja jest tak prosta jak w wypadku wspomnianych spółników. Oto garść przykładów:

Gdy zdecydujemy się zaprzestać otaczać polecenia ramkami, to do zmiany wyglądu wystarczy, że zmienimy definicję otoczenia `command`. Jest to znacznie łatwiejsze od przebiegnięcia przez cały dokument w celu wylapania w nim wszystkich standardowych poleceń L^AT_EX, które służą do rysowania ramek wokół słów.

We wstępie do tego opracowania wspomnieliśmy, że w L^AT_EXu możemy się skupić na logicznej strukturze dokumentów. Wskazane jest rozróżnić w tekście źródłowym wszystkie elementy logiczne dokumentu, nawet jeżeli ich formatowanie jest identyczne. Nierzadko bowiem to, co dzisiaj formatujemy w taki sam sposób, w przyszłości możemy chcieć rozróżnić.

Zwyczaj się na przykład składać adresy internetowe imitacją kroju maszynowego. Ponieważ w adresach mogą wystąpić znaki specjalne L^AT_EXa, to można by do tego celu stosować instrukcję `\verb`. Lepiej jednak użyć specjalnej instrukcji, np. `\url`. W dokumencie papierowym nie ma to znaczenia, znakowanie logiczne pozwala jednak przedstawić dokument zarówno w formie drukowanej, jak i hipertekstowej, w formacie HTML lub PDF.

6.1.1. Instrukcje definiowane przez użytkownika

Do definiowania potrzebnych nam nowych poleceń możemy użyć instrukcji:

```
\newcommand{nazwa}[num]{tekst}
```

Wymaga ona podania dwóch argumentów. Pierwszy z nich, *nazwa*, oznacza nazwę nowej instrukcji, natomiast *tekst* to jej znaczenie, czyli tekst, który ma zostać wstawiony do składu w momencie wykonania instrukcji. Podawany w nawiasach kwadratowych argument *num* powinien być cyfrą od 1 do 9, określającą liczbę (obowiązkowych) argumentów instrukcji. Argument *num* jest opcjonalny, a jego pominięcie oznacza, że definiowana instrukcja jest bezargumentowa.

W części *tekst* wolno używać zarówno standardowych instrukcji L^AT_EXa, jak też zdefiniowanych przez użytkownika. Nie wolno jednak korzystać z tych instrukcji, które same definiują inne polecenia, jak `\newcommand`, `\newenvironment` itp. Niedozwolona jest rekursja, nie wolno też w nazwach instrukcji umieszczać polskich liter diakrytycznych.

Następujące przykłady pomogą lepiej zrozumieć zagadnienie. W pierwszym z nich definiujemy instrukcję o nazwie `\kwle`, mającą być skrótem dla słów „Krótkie wprowadzenie do systemu L^AT_EX 2_ε”. Takie polecenie mogłoby się przydać, gdyby tytuł książki miał w niej występować wielokrotnie.

```
\newcommand{\kwle}{Krótkie
wprowadzenie do systemu \LaTeXe}
% następnie po \begin{document}:
\kwle; \emph{\kwle}
```

```
Krótkie wprowadzenie do systemu LATEX 2ε;
Krótkie wprowadzenie do systemu LATEX 2ε
```

Pakiet `marvosym` dostarcza wielu różnych symboli, w tym euro pod nazwą `\EURtm`. Jego wadą jest to, że nie udostępnia wersji pochylonej i wytłuszczonej euro.

Tabela 2.1: Torba pełna symboli euro

LM+textcomp	<code>\texteuro</code>	€	€	€
eurosym	<code>\euro</code>	€	€	€
[gen]eurosym	<code>\euro</code>	€	€	€
marvosym	<code>\EURtm</code>	€	€	€

2.4.7. Wielokropek (...)

W typowym piśmie maszynowym przecinek oraz kropka zajmują tyle samo miejsca co każdy inny znak. W piśmie drukarskim szerokość tych znaków jest z reguły bardzo mała i dlatego, jeżeli umieścimy je obok siebie, to odstępy między nimi będą zbyt małe. Do uzyskiwania wielokropka (trzech kropek) używamy instrukcji `\ldots`. Przykład:

`\ldots`

Nie tak ..., lecz raczej tak:
Nowy Jork, Tokio, Budapeszt, `\ldots`

Nie tak ..., lecz raczej tak:
Nowy Jork, Tokio, Budapeszt, ...

2.4.8. Ligatury

Ligatury (spójki) to znaki graficzne, w których połączono dwie lub trzy litery. W niektórych językach ligatury występują jako właściwe danej ortografii znaki pisma, np. *œ* w języku francuskim. Większość ligatur tworzy się ze względów estetycznych lub zwyczajowych. \LaTeX zna pięć następujących ligatur:

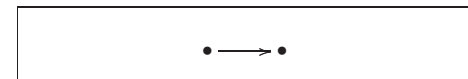
`ff fi fl ffi ffl` zamiast `ff fi fl ffi ffl`

\TeX używa ligatur automatycznie. Można temu zapobiec, między znakami tworzącymi ligaturę wstawiając instrukcję `\mbox{}`:

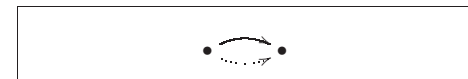
Jak lepiej: geografii czy
`geograf\mbox{ }ii?`

Jak lepiej: geografii czy geografii?

```
\begin{displaymath}
\ymatrix{
\bullet \ar[r] & \bullet \\
\bullet \ar@{.}>[r] & \bullet
}
\end{displaymath}
```



```
\begin{displaymath}
\ymatrix{
\bullet \ar@/^/[r] & \bullet \\
\bullet \ar@/_/@{.}>[r] & \bullet
}
\end{displaymath}
```



Symbole między ukośnikami określają tu sposób rysowania krzywych.

Pakiet `Xy-pic` oferuje wiele metod wpływania na sposób rysowania krzywych. Więcej na ten temat można przeczytać w dokumentacji pakietu [23].

3. Należy włączyć specyficzne dla danego języka reguły typograficzne. Na przykład w języku francuskim każdy dwukropek i wykrzyknik trzeba poprzedzić odstępem, a w języku polskim po numerach w tytułach rozdziałów i punktów stawia się kropkę.

Jeżeli dysponujemy dobrze skonfigurowanym \LaTeX em, czyli \LaTeX em z wygenerowanym formatem zawierającym odpowiednie reguły przenoszenia wyrazów, to resztę zadań polonizacyjnych załatwi pakiet `babel`. Wystarczy w tym celu do preambuły dokumentu wpisać instrukcję:

```
\usepackage{polish}
```

Jej ogólną postacią jest:

```
\usepackage[lista-języków]{babel}
```

Argument `lista-języków` to oddzielone przecinkami nazwy języków, które obsługuje zainstalowana wersja \LaTeX a. Ostatni na liście jest językiem domyślnym. Do przełączenia się w treści dokumentu na inny język służy polecenie:

```
\selectlanguage{język}
```

Jeśli używana wersja \LaTeX a nie obsługuje języka z listy, to `babel` zadziała z wyłączonym przenoszeniem wyrazów, co znacznie pogorszy jakość składu.

Pakiet `babel` dla każdego języka definiuje elementy wpisywane automatycznie przez program (np. dla języka polskiego „Spis treści” zamiast „Table of contents”) oraz udostępnia polecenia ułatwiające przygotowanie dokumentów w tym języku.

Dla niektórych języków `babel` udostępnia specjalne instrukcje, ułatwiające wprowadzanie znaków diakrytycznych i specjalnych. Teksty w języku niemieckim zawierają na przykład sporo liter z umlautami: (äöü). Wykorzystując pakiet `babel`, można wprowadzić literę ö, wpisując `"o` zamiast `\"o`.

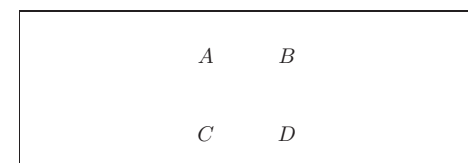
W większości systemów komputerowych znaki akcentowane i specjalne (czyli znaki o kodach ASCII większych niż 127) można wprowadzać bezpośrednio z klawiatury. Przykładowo, polskie znaki diakrytyczne można wprowadzać, naciskając klawisz prawy-Alt i jednocześnie klawisz z odpowiednią literą. \LaTeX radzi sobie z takimi znakami. Począwszy od grudnia 1994 r. dystrybucje \LaTeX a zawierają pakiet `inputenc`, pozwalający kodować znaki diakrytyczne w różnych wariantach. Przykładowo, jeśli dokument jest kodowany w standardzie ISO 8859-2 (system operacyjny Unix/Linux), to pakiet `inputenc` należy dołączyć do dokumentu w następujący sposób:

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

W wypadku dokumentu kodowanego w standardzie CP 1250 (system operacyjny MS Windows) powinniśmy zamiast opcji `latin2` wpisać `cp1250`. Dla dokumentów unikodowych należy użyć opcji `utf8`.

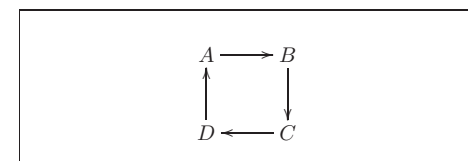
Chociaż pakiety `babel` oraz `inputenc` umożliwiają skład dokumentów w języku polskim, to nie są pozbawione wad. Kłopoty mogą sprawiać dokumenty

```
\begin{displaymath}
\matrix{A & B \\
C & D }
\end{displaymath}
```



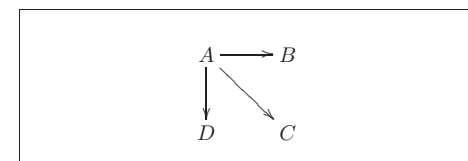
Polecenia `\matrix` można użyć jedynie w trybie matematycznym. Powyżej zadano dwa wiersze i dwie kolumny. Aby przekształcić tę macierz w graf, posługujemy się dodającym strzałki poleceniem `\ar`⁴:

```
\begin{displaymath}
\matrix{ A \ar[r] & B \ar[d] \\
D \ar[u] & C \ar[l] }
\end{displaymath}
```



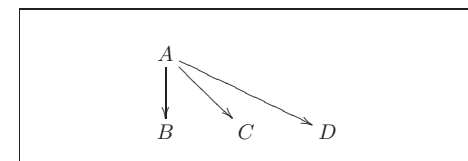
Polecenie `\ar` umieszczamy w komórce, w której strzałka ma się zaczynać. Argument oznacza kierunek strzałki (u – góra, d – dół, r – prawo i l – lewo):

```
\begin{displaymath}
\matrix{
A \ar[d] \ar[dr] \ar[r] & B \\
D & & C }
\end{displaymath}
```



Aby uzyskać przekątne, używamy po prostu więcej niż jednego kierunku. Powtarzając kierunek, wydłużamy w istocie strzałkę:

```
\begin{displaymath}
\matrix{
A \ar[d] \ar[dr] \ar[dr] & & \\
B & & C & D }
\end{displaymath}
```



Uzupełniając strzałki o etykiety, możemy uzyskać jeszcze ciekawsze grafy. Używamy do tego zwyczajnych operatorów indeksów górnych i dolnych:

⁴Pakiety `Xy-pic` oraz `polish` są w konflikcie, bo oba definiują polecenie `\ar`. Jeżeli polecenie `\ar` ma działać tak, jak zdefiniowano to w pakiecie `Xy-pic`, to trzeba xy dołączyć po pakiecie `polish`. Oczywiście nie ma wtedy dostępu do polecenia `\ar` z pakietu `polish`.

```
\documentclass{article}
\usepackage[polish]{babel}
\usepackage[cp1250]{inputenc}
\usepackage[OT4]{fontenc} %% lub [T1]
```

Lepszy sposób polonizacji \LaTeX a opisano w następnym punkcie.

Rodzina fontów LH zawiera litery potrzebne do składania dokumentów w cyrylicy. Ze względu na dużą liczbę znaków w różnych pismach cyrylickich są one zgrupowane w czterech różnych kodowaniach: T2A, T2B, T2C, i X2⁴. Rodzina CB, w kodowaniu LGR, zawiera fonty do składu greki.

2.5.1. Język polski w dokumentach

Jak wspomniano w punkcie 2.5, poprawny skład w języku polskim wymaga trzech rzeczy: \LaTeX a z wbudowanymi w format polskimi regułami przenoszenia wyrazów, fontów zawierających polskie znaki diakrytyczne oraz dodatkowego pakietu obsługującego specyficzne dla języka polskiego reguły typograficzne. W tym punkcie zakładamy, że udało ci się skonfigurować \LaTeX a pod kątem pierwszych dwóch punktów, tj. wygenerowania formatu i zainstalowania fontów PL lub EC, i koncentrujemy się na szczegółowym opisie ostatniego aspektu polonizacji.

W każdej dystrybucji \LaTeX a znajduje się wspomniany w punkcie 2.5 pakiet babel. Mimo występujących w nim niedociągnięć można go polecić, szczególnie początkującym. Bardziej wymagający użytkownicy piszący po polsku powinni korzystać z opisanego dalej zestawu polonizacyjnego platex.

W wypadku języka polskiego wiele kłopotów sprawia kodowanie znaków diakrytycznych. Wynika to przede wszystkim z braku standardu: różne platformy systemowe promują w tym zakresie różne rozwiązania⁵. Można wyróżnić dwa sposoby zapisu diakrytyków w \LaTeX u: polecenia standardowe (opisane w punkcie 2.4.9 oraz poniżej) i notację „bezpośrednią”, posługującą się znakami o kodach większych od 127⁶.

Standardowe polecenia akcentowe \LaTeX a umożliwiają zapis wszystkich polskich znaków diakrytycznych w następującej postaci⁷:

⁴Listę języków obsługiwanych w tych kodowaniach można znaleźć w [26].

⁵Lekarstwem na tę bolączkę może być kodowanie wielobajtowe, tj. standard Unicode (UTF). Wprawdzie \LaTeX potrafi przetwarzać dokumenty unikodowe jedynie w ograniczonym zakresie, ale jest on wystarczający dla tekstów w językach europejskich. Specjalna wersja \TeX a, XeTeX autorstwa Jonathana Kew, potrafi przetwarzać dokumenty zakodowane w unikodzie, a także generować skład z użyciem unikodowych fontów OpenType.

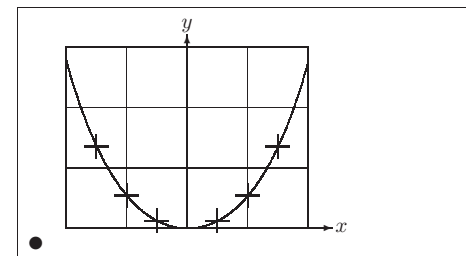
⁶Jest jeszcze trzeci sposób: tak zwana notacja prefiksowa ($/\mathbf{a} - \mathbf{q}$, $/\mathbf{n} - \mathbf{ú}$, $/\mathbf{S} - \mathbf{Š}$ itp.). Metoda ta wyszła już w zasadzie z użytku, gdyż powszechnie dostępna jest metoda bezpośredniego wprowadzania znaków z polskimi ogonkami z klawiatury.

⁷Z notacją tą wiąże się jednak pewien problem: otoczenie `\tapping` zmienia lokalnie definicję kilku makr, w tym `\'`. Dlatego w jego obrębie do uzyskiwania znaków z akcentem *acute* trzeba używać notacji typu `\a'o`. Konsekwencją jest to, że zarówno w implementacji notacji prefiksowej, jak i „stron kodowych” pakietu `inputenc` trzeba się do akcentu *acute* dostawać nieco naokolo.

W [21] można znaleźć program w Javie, który generuje odpowiednią linię polecenia `\qbezier`.

5.2.10. Krzywe łańcuchowe

```
\setlength{\unitlength}{1cm}
\begin{picture}(4.3,3.6)(-2.5,-0.25)
\put(-2,0){\vector(1,0){4.4}}
\put(2.45,-.05){\mathbf{x}}
\put(0,0){\vector(0,1){3.2}}
\put(0,3.35){\makebox(0,0){\mathbf{y}}}
\qbezier(0.0,0.0)(1.2384,0.0)
(2.0,2.7622)
\qbezier(0.0,0.0)(-1.2384,0.0)
(-2.0,2.7622)
\linethickness{.075mm}
\multiput(-2,0)(1,0){5}
{\line(0,1){3}}
\multiput(-2,0)(0,1){4}
{\line(1,0){4}}
\linethickness{.2mm}
\put(.3,.12763){\line(1,0){.4}}
\put(.5,-.07237){\line(0,1){.4}}
\put(-.7,.12763){\line(1,0){.4}}
\put(-.5,-.07237){\line(0,1){.4}}
\put(.8,.54308){\line(1,0){.4}}
\put(1,.34308){\line(0,1){.4}}
\put(-1.2,.54308){\line(1,0){.4}}
\put(-1,.34308){\line(0,1){.4}}
\put(1.3,1.35241){\line(1,0){.4}}
\put(1.5,1.15241){\line(0,1){.4}}
\put(-1.7,1.35241){\line(1,0){.4}}
\put(-1.5,1.15241){\line(0,1){.4}}
\put(-2.5,-0.25){\circle*{0.2}}
\end{picture}
```



Każdą z symetrycznych połówek wykresu cosinusa hiperbolicznego $y = \cosh x - 1$ przybliżono na rysunku krzywą Béziera. Prawa połówka krzywej kończy się w punkcie $(2, 2.7622)$, w którym nachylenie ma wartość $m = 3.6269$. Używając ponownie równania (5.1), możemy wyliczyć pośrednie punkty kontrolne. Okazuje się, że są to: $(1.2384, 0)$ i $(-1.2384, 0)$. Krzyżkami została zaznaczona „prawdziwa” krzywa. Błąd jest ledwie zauważalny, bo wynosi mniej niż jeden procent.

Ten przykład ilustruje też użycie opcjonalnego argumentu otoczenia `\begin{picture}`. Rysunek zdefiniowano w terminach wygodnych współrzędnych „matematycznych”, podczas gdy poleceniem:

```
\begin{picture}(4.3,3.6)(-2.5,-0.25)
```

nomathsymbols blokada zmiany znaczenia standardowych poleceń trygonometrycznych oraz symboli relacji mniejszy-lub-równy i większy-lub-równy (por. punkt 3.10, str. 59);

MeX tryb pełnej polonizacji (zalecane).

Jeżeli opcję układu kodowania w foncie pominięto, to pakiet polski używa fontów PL (w wypadku ich braku L^AT_EX będzie sygnalizował błędy). Dotyczy to zarówno fontów tekstowych, jak i zawierających znaki matematyczne. W instalacji zawierającej fonty PL dołączenie pakietu polski bez opcji jest równoważne poleceniu:

```
\usepackage[OT4,plmath]{polski}
```

Polecenie `\selecthyphenation` pozwala przełączyć się na dany zestaw wzorców dzielenia wyrazów. Jest to odpowiednik polecenia `\selectlanguage` z pakietu babel. Argumentem jest nazwa języka.

Polonizacyjnym uzupełnieniem pakietu polski jest zestaw klas Marcina Wolińskiego `mwart`, `mwrep` i `mwbook`, dostępny pod adresem <http://www.mimuw.edu.pl/~wolinski/mwcls.html>. W klasach tych zostały uwzględnione m.in. takie zwyczaje jak: umieszczanie kropek po numerach śródtytułów, sposób formatowania przypisów oraz pagin, zakaz przenoszenia słów w śródtytułach, reguły umieszczania/pomijania paginacji. Uzyskano to za cenę znacznej ingerencji w sposób działania klas standardowych, czego skutkiem jest niekompatybilność z częścią pakietów L^AT_EXa.

Oto przykładowa preambuła artykułu składanego w klasie `mwart`:

```
%& --translate-file=cp1250pl
\documentclass{mwart}
\usepackage[MeX]{polski}
\begin{document} ...
```

Jest to zalecany sposób rozpoczynania dokumentów w języku polskim. Warto z niego korzystać, zaopatrzywszy się w niezbędne elementy: fonty PL, pakiet polski i klasy Marcina Wolińskiego. W standardowej dystrybucji L^AT_EXa, w której na ogół znajdują się obecnie fonty PL, zadziała natomiast taka preambuła:

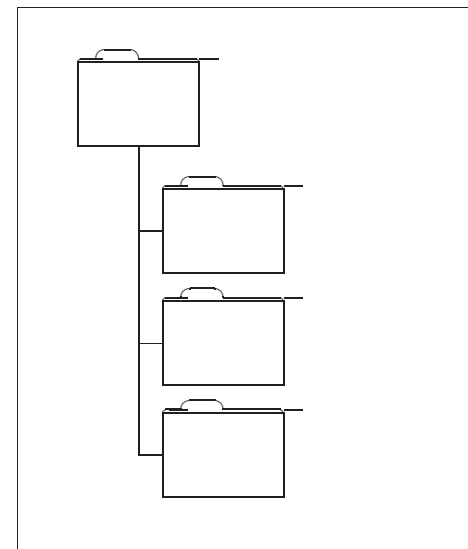
```
%& --translate-file=cp1250pl
\documentclass{article}
\usepackage{polski}
```

2.6. Odstęp między wyrazami

Aby wyrównać prawy margines, L^AT_EX wstawia między słowami odstępy różnej wielkości. Odstęp wstawiany na końcu zdania jest trochę większy, ponieważ tak składa się książki w krajach anglosaskich. L^AT_EX zakłada, że zdania mogą się kończyć kropką, znakiem zapytania lub wykrzyknikiem.

5.2.8. Wielokrotne użycie pudełek z rysunkami

```
\setlength{\unitlength}{0.5mm}
\begin{picture}(120,168)
\newsavebox{foldera}
\savebox{foldera}
(40,32)[bl]{% definition
\multiput(0,0)(0,28){2}
{\line(1,0){40}}
\multiput(0,0)(40,0){2}
{\line(0,1){28}}
\put(1,28){\oval(2,2)[tl]}
\put(1,29){\line(1,0){5}}
\put(9,29){\oval(6,6)[tl]}
\put(9,32){\line(1,0){8}}
\put(17,29){\oval(6,6)[tr]}
\put(20,29){\line(1,0){19}}
\put(39,28){\oval(2,2)[tr]}
}
\newsavebox{folderb}
\savebox{folderb}
(40,32)[l]{% definition
\put(0,14){\line(1,0){8}}
\put(8,0){\usebox{foldera}}
}
\put(34,26){\line(0,1){102}}
\put(14,128){\usebox{foldera}}
\multiput(34,86)(0,-37){3}
{\usebox{folderb}}
\end{picture}
```



Pudełko rysunku można *zadeklarować* instrukcją:

```
\newsavebox{nazwa}
```

następnie *zachować* poleceniem:

```
\savebox{nazwa}(szerokość,wysokość)[pozycja]{treść}
```

i ostatecznie dowolnie często *rysować*, wywołując:

```
\put(x,y)\usebox{nazwa}
```

Opcjonalny parametr *pozycja* definiuje „punkt zaczepienia” zachowywanego pudełka (`savebox`). W przykładzie nadano mu wartość `bl`, co oznacza umieszczenie punktu zaczepienia w dolnym lewym narożniku pudełka. Pozycje można też oznaczać literami `t` (górze) i `r` (prawo).

- instrukcja `\part` nie ma wpływu na numerację rozdziałów;
- instrukcja `\appendix` nie ma argumentów. Jest to deklaracja zmieniająca sposób numerowania z cyfr na litery. Dotyczy to rozdziałów w klasach `book` i `report`, a punktów w klasie `article`.

Argumentu instrukcji podziału dokumentu \LaTeX używa do przygotowania spisu treści. Instrukcja:

```
\tableofcontents
```

wstawia spis treści w miejscu jej użycia. Aby w spisie treści otrzymać poprawne numery stron, trzeba dokument przetworzyć („złatechować”) dwukrotnie. Czasami niezbędna jest nawet trzecia kompilacja. Kolejny przebieg jest potrzebny, gdy pod koniec przetwarzania dokumentu \LaTeX pokazuje komunikat:

LaTeX Warning: Label(s) may have changed.

Rerun to get cross-references right.

\LaTeX przetwarza dokument strona po stronie i w pojedynczym przebiegu nie może wstawić spisu treści na początku dokumentu, ponieważ nie jest jeszcze znana jego treść ani numeracja stron. Podobnie ma się sprawa ze spisami tabel czy rysunków. Rozwiązanie tego problemu jest tyleż proste co skuteczne. Podczas przetwarzania dokumentu \LaTeX zapisuje odpowiednie informacje do plików pomocniczych – w celu ich wykorzystania podczas kolejnych przebiegów.

Przeznaczenie danego pliku pomocniczego jest sygnalizowane przez rozszerzenie jego nazwy. I tak: plik o rozszerzeniu `.toc` zawiera spis treści, plik `.lot` – spis tabel, plik `.lof` – spis rysunków, `.aux` – informacje o odsyłaczach wewnątrz dokumentu (odsyłacze omawiamy w punkcie 2.8). Pełniejszy wykaz nazw plików \LaTeX owych podano w punkcie 1.7 na stronie 12.

Wymienione wyżej instrukcje podziału hierarchicznego posiadają także wersje „z gwiazdką”. Nazwa instrukcji w wersji „z gwiazdką” składa się z „normalnej” nazwy, po której występuje znak „*”. W wyniku działania takiej instrukcji tytuł rozdziału lub punktu zostanie umieszczony w dokumencie, ale nie w spisie treści; tytuł nie zostanie też objęty numeracją. Przykładowo, wersją „z gwiazdką” instrukcji `\section{Pomoc}` jest `\section*{Pomoc}`.

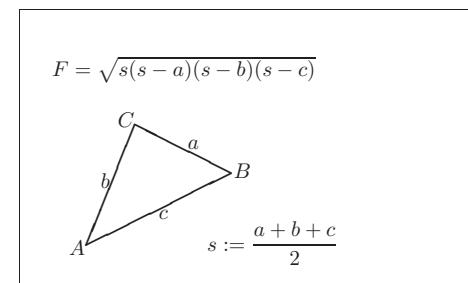
Zwyczajem angielskim jest nieumieszczanie nienumerowanych śródtytułów w spisie treści. Polscy redaktorzy często się domagają, by na przykład „Wstęp” był śródtytułem nienumerowanym, a jednocześnie występował w spisie. Ten problem rozwiązujemy za pomocą polecenia:

```
\addcontentsline{spis}{poziom}{śródtytuł}
```

gdzie: *spis* to rozszerzenie nazwy pliku, w którym ma zostać zapisana informacja, *poziom* to `chapter`, `section` bądź inna nazwa polecenia hierarchicznego, a *śródtytuł* to sam śródtytuł. Przykład:

5.2.5. Tekst i wzory

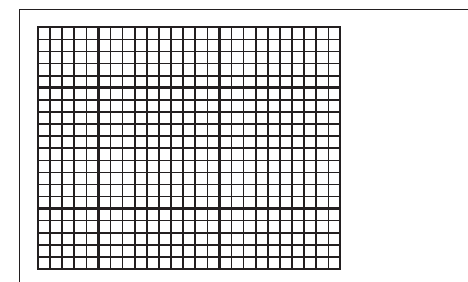
```
\setlength{\unitlength}{0.8cm}
\begin{picture}(6,5)
  \thicklines
  \put(1,0.5){\line(2,1){3}}
  \put(4,2){\line(-2,1){2}}
  \put(2,3){\line(-2,-5){1}}
  \put(0.65,0.3){$A$}
  \put(4.05,1.9){$B$}
  \put(1.65,2.95){$C$}
  \put(3.1,2.5){$a$}
  \put(1.3,1.7){$b$}
  \put(2.5,1.05){$c$}
  \put(0.3,4){$F=$}
  \sqrt{s(s-a)(s-b)(s-c)}$}
  \put(3.5,0.4){$\displaystyle
s:=\frac{a+b+c}{2}$}
\end{picture}
```



Jak widać w powyższym przykładzie, tekst oraz wzory można łatwo wprowadzać do otoczenia `picture` poleceniem `\put`.

5.2.6. Polecenia \multiput i \linethickness

```
\setlength{\unitlength}{2mm}
\begin{picture}(30,20)
  \linethickness{0.075mm}
  \multiput(0,0)(1,0){26}%
  {\line(0,1){20}}
  \multiput(0,0)(0,1){21}%
  {\line(1,0){25}}
  \linethickness{0.15mm}
  \multiput(0,0)(5,0){6}%
  {\line(0,1){20}}
  \multiput(0,0)(0,5){5}%
  {\line(1,0){25}}
  \linethickness{0.3mm}
  \multiput(5,0)(10,0){2}%
  {\line(0,1){20}}
  \multiput(0,5)(0,10){2}%
  {\line(1,0){25}}
\end{picture}
```



Polecenie:

```
\multiput(x,y)(\Delta x,\Delta y){n}{object}
```

ma cztery argumenty: punkt początkowy, wektor przesunięcia z jednego obiektu do kolejnego, liczbę obiektów oraz rysowany obiekt. Instrukcja

2.8. Odsyłacze

Książki, raporty i artykuły często zawierają odsyłacze do rysunków, tabel i innych fragmentów tekstu. Z odsyłaczami związane są w \LaTeX u następujące trzy instrukcje:

```
\label{etykieta}, \ref{etykieta} i \pageref{etykieta}
```

Argument *etykieta* jest ciągiem liter, cyfr lub znaków interpunkcyjnych. Nazwy etykiet ustala sam autor. \LaTeX zamienia $\ref{etykieta}$ na numer tego rozdziału, punktu, rysunku, tabeli czy też równania matematycznego, *bezpośrednio* za którym umieszczona została instrukcja \label zawierająca identyczną *etykieta*. Instrukcja $\pageref{etykieta}$ działa identycznie jak \ref , z tym że wstawia numer strony, na której znajduje się element oznaczony *etykieta*¹⁰. Oto przykład:

Odsyłacz do tego punktu

```
\label{sec:this} wygląda tak:
,,patrz punkt~\ref{sec:this} na
stronie~\pageref{sec:this}.''
```

```
Odsyłacz do tego punktu wygląda tak: „patrz
punkt 2.8 na stronie 34.”
```

Podobnie jak w wypadku spisów treści, tabel czy rysunków, do ustalenia właściwej numeracji odsyłaczy potrzebne są co najmniej dwie, a z reguły trzy kompilacje dokumentu. Podczas pierwszej \LaTeX wysyła do pliku pomocniczego z rozszerzeniem `.aux` (zob. punkt 1.7) informacje o odsyłaczach, które wykorzystuje podczas kolejnych kompilacji.

2.9. Przypisy

Do składania przypisów u dołu strony służy instrukcja:

```
\footnote{tekst przypisu}
```

Należy ją wstawić bezpośrednio po słowie lub zdaniu, do którego się odnosi. W krajach anglosaskich przypisy odnoszące się do całego zdania lub jego części umieszczają się natychmiast po kropce lub przecinku. W Polsce najczęściej umieszczają się *przed* znakiem przestankowym (zasadę tę stosujemy w niniejszym tłumaczeniu).

\LaTeX numeruje przypisy automatycznie. Sposób ich numerowania zależy od używanej klasy. W klasie `article` numeracja jest ciągła, w klasach `report` i `book` przypisy są numerowane w ramach rozdziałów.

¹⁰Warto pamiętać, że te instrukcje „nie wiedzą”, do czego tak naprawę się odnoszą. Zadaniem instrukcji \label jest przechowanie związku wygenerowanej automatycznie liczby z miejscem w tekście.

Do rysowania odcinków służy polecenie:

```
\put(x,y){\line(x1,y1){length}}
```

Ma ono dwa argumenty: wektor kierunku i długość.

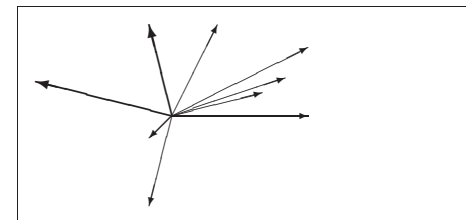
Jako składników wektora kierunku można użyć jedynie liczb całkowitych:

$-6, -5, \dots, 5, 6,$

Muszą one na dodatek być liczbami względnie pierwszymi (jedynym ich wspólnym dzielnikiem może być 1). Powyższy rysunek ilustruje wszystkie możliwe wartości nachylenia w pierwszej ćwiartce płaszczyzny. Długość jest podawana w jednostkach \unitlength . Argument długości oznacza współrzędną pionową w wypadku odcinka pionowego, zaś współrzędną poziomą we wszystkich pozostałych przypadkach.

5.2.3. Strzałki

```
\setlength{\unitlength}{0.75mm}
\begin{picture}(60,40)
\put(30,20){\vector(1,0){30}}
\put(30,20){\vector(4,1){20}}
\put(30,20){\vector(3,1){25}}
\put(30,20){\vector(2,1){30}}
\put(30,20){\vector(1,2){10}}
\thicklines
\put(30,20){\vector(-4,1){30}}
\put(30,20){\vector(-1,4){5}}
\thinlines
\put(30,20){\vector(-1,-1){5}}
\put(30,20){\vector(-1,-4){5}}
\end{picture}
```



Strzałki rysuje się poleceniem:

```
\put(x,y){\vector(x1,y1){length}}
```

W wypadku strzałek na składniki wektora kierunku nałożone są jeszcze większe ograniczenia niż dla odcinków, bo jedynymi dopuszczalnymi liczbami całkowitymi są:

$-4, -3, \dots, 3, 4.$

Wartości składników i tu muszą być liczbami względnie pierwszymi (jedynym wspólnym dzielnikiem może być 1). Zauważ efekt działania polecenia \thicklines na dwie strzałki skierowane w stronę lewego górnego narożnika oraz \thinlines na strzałki w stronę narożnika prawego górnego.

2.11.1. Otoczenia `itemize`, `enumerate` i `description`

Otoczenia `itemize` oraz `description` służą do tworzenia wyszczególnień, zaś `enumerate` do tworzenia wyliczeń. W każdym z nich element wyliczenia zaczyna się od instrukcji `\item`.

```
\begin{enumerate}
\item Taka lista:
\begin{itemize}
\item wygląda
\item[--] śmiesznie.
\end{itemize}
\item Pamiętaj:
\begin{description}
\item[Głupoty] nie staną się mądrościami,
    gdy się je wyliczy.
\item[Mądrości] można elegancko
    zestawiać w wyliczeniach.
\end{description}
\end{enumerate}
```

- | |
|---|
| <p>1. Taka lista:</p> <ul style="list-style-type: none"> – wygląda – śmiesznie. <p>2. Pamiętaj:</p> <p>Głupoty nie staną się mądrościami, gdy się je wyliczy.</p> <p>Mądrości można elegancko zestawiać w wyliczeniach.</p> |
|---|

2.11.2. Otoczenia `flushleft`, `flushright` i `center`

W otoczeniach `flushleft` i `flushright` akapity są składane z wyrównaniem, odpowiednio, do lewego bądź prawego marginesu. Wewnątrz otoczenia `center` każdy wiersz akapitu jest wyśrodkowany w osi szpalty. Tak jak zawsze, \LaTeX dzieli akapity na wiersze automatycznie, można jednak w obrębie powyższych otoczeń wymusić zmianę wiersza poleceniem `\\`.

```
\begin{flushleft}
To jest tekst\\ wyrównany do lewej.
{\LaTeX} nie składa tu wierszy\\
z zachowaniem jednakowej długości.
\end{flushleft}
```

<p>To jest tekst wyrównany do lewej. \LaTeX nie składa tu wierszy z zachowaniem jednakowej długości.</p>

```
\begin{flushright}
To jest tekst\\ wyrównany do prawej.
{\LaTeX} nie składa tu wierszy\\
z zachowaniem jednakowej długości.
\end{flushright}
```

<p>To jest tekst wyrównany do prawej. \LaTeX nie składa tu wierszy z zachowaniem jednakowej długości.</p>
--

```
\begin{center}
To jest tekst\\wyśrodkowany.
\end{center}
```

<p>To jest tekst wyśrodkowany.</p>
--

Pakiety takie jak: `epic` i `eepic` (opisane na przykład w [6]) bądź `pstricks` pomagają pokonywać ograniczenia krępujące oryginalne otoczenie `picture` i znacznie rozszerzają graficzną moc \LaTeX a.

O ile pierwsze dwa z tych pakietów jedynie rozszerzają otoczenie `picture`, to pakiet `pstricks` zawiera własne otoczenie rysujące – `pspicture`. Moc pakietu `pstricks` bierze się z tego, że istotnie wykorzystuje on możliwości Postscriptu.

Na dodatek napisano wiele pakietów realizujących konkretne cele. Jednym z nich jest `Xy-pic`, opisany na końcu tego rozdziału. Znaczną liczbę tych pakietów opisano w [7] (nie należy mylić z [6]).

Spośród narzędzi graficznych związanych z \LaTeX em największe chyba możliwości posiada `METAPOST`, autorstwa Johna D. Hobby, będący wariantem programu `METAFONT`, którego autorem jest z kolei Donald E. Knuth. `METAPOST` jest wyposażony w solidny i matematycznie wyrafinowany język programowania `METAFONT`a. Inaczej jednak niż `METAFONT`, który generuje mapy bitowe, `METAPOST` generuje pliki Postscriptowe, które można importować do \LaTeX a. Za wprowadzenie może posłużyć [8], a za podręcznik – [21]. Dostępne jest także wprowadzenie do programu `METAPOST` w języku polskim [8]².

Obszerne omówienie zagadnień związanych z wykorzystaniem grafiki (oraz fontów) w \LaTeX u i \TeX u można znaleźć w [9].

5.2. Otoczenie `picture`

Urs Oswald <osurs@bluewin.ch>

5.2.1. Podstawowe polecenia

Otoczenie `picture`³ można tworzyć poleceniem:

```
\begin{picture}(x,y)...\end{picture}
```

lub

```
\begin{picture}(x,y)(x_0,y_0)...\end{picture}
```

Liczby x , y , x_0 , y_0 odnoszą się do wielkości `\unitlength`, którą można zmieniać w dowolnym momencie (jednak nie wewnątrz otoczenia `picture`) poleceniem takim jak:

```
\setlength{\unitlength}{1.2cm}
```

²Nie ma co ukrywać, że posługiwanie się `METAPOST`em wymaga posiadania przynajmniej minimum umiejętności programistycznych. Dla użytkowników którzy nie programują przydatny może być program `MetaGraf`, który jest okienkową nakładką do programu `METAPOST`, por. <http://w3.mecanica.upm.es/metapost/>.

³Otoczenie `picture` działa w standardowym \LaTeX 2_ε; nie trzeba łączyć żadnych dodatkowych pakietów.

2.11.5. Symulacja maszynopisu

Tekst zawarty między `\begin{verbatim}` a `\end{verbatim}` jest przez \LaTeX a składany dosłownie, czyli tak, by wyglądał jak napisany na maszynie, z zachowaniem zmian wiersza i odstępów z pliku źródłowego.

Aby uzyskać ten efekt, krój pisma zmienia się na imitujący pismo maszynowe (grotesk). Wszystkie znaki w tym kroju, włączając spację, mają jednakową szerokość. Zakończenie linii wewnątrz otoczenia `verbatim` prowadzi do rozpoczęcia nowego wiersza w wydruku, a *każda* spacja zamienia się na odstęp. Wewnątrz otoczenia `verbatim` nie są wykonywane instrukcje.

Wewnątrz akapitów imitację maszynopisu uzyskuje się za pomocą instrukcji:

```
\verb+tekst+
```

Znak `+` ogranicza tekst, który ma zostać wydrukowany dosłownie. Zamiast `+` można użyć innego znaku, byle to nie była litera, gwiazdka, spacja ani żaden znak, który występuje w *tekście*. Instrukcję `\verb` oraz otoczenie `verbatim` wykorzystujemy często w tej książce do składania przykładów \LaTeX owych.

Rozważmy przykład `\ldots`

```
\begin{verbatim}
{ for (i=1;i<=NF;i++) {l[$i]++; }
END {for (i in l) {print l[i]}
\end{verbatim}
```

Rozważmy przykład...

```
{ for (i=1;i<=NF;i++) {l[$i]++; }
END {for (i in l) {print l[i]}
```

Otoczenie `verbatim` oraz instrukcja `\verb` mają także wersje „z gwiazdką”, w których spacja z pliku źródłowego jest zamieniana na znak `_`. Jest to jedyna różnica działania w porównaniu do wersji bezgwiazdkowych:

```
\begin{verbatim*}
gwiazdkowa wersja
otoczenia verbatim
wyróżnia spacje
w tekście
\end{verbatim*}
```

```
gwiazdkowa_wersja
otoczenia_verbatim
wyróżnia_spacje
w_tekście
```

Otoczenia `verbatim` ani instrukcji `\verb` nie wolno używać wewnątrz argumentów innych instrukcji (więcej na ten temat w punkcie 2.13).

2.11.6. Otoczenie tabular

Do składania tabel służy otoczenie `tabular`. \LaTeX automatycznie ustala szerokość poszczególnych rubryk tabeli. Otoczenie ma jeden parametr obowiązkowy, *spec-kolumn*, który określa liczbę kolumn tabeli oraz sposób ich justowania:

```
\begin{tabular}{spec-kolumn}
```

trybów `\mode<trans>` i `\mode<handout>` nie zadano żadnej opcji; mają się ukazywać w swoim układzie domyślnym.

Polecenia `\title`, `\author`, `\institute`, oraz `\titlegraphic` określają zawartość strony tytułowej. Opcjonalne argumenty instrukcji `\title` i `\author` pozwalają podać specjalną wersję tytułu oraz autora do wyświetlenia w panelu tematu *Goettingen*. Tytuły oraz podtytuły na panelu są tworzone przez zwykłe polecenia `\section` i `\subsection`, umieszczone *poza* otoczeniem `frame`.

Małe ikony u dołu ekranu służą do nawigowania po dokumencie. Można ich ukazywanie się zablokować poleceniem:

```
\setbeamertemplate{navigation symbols}{}
```

Zawartość każdego slajdu oraz ekranu należy umieścić w otoczeniu `frame`. W nawiasach kątowych `< i >` można podać opcjonalny argument, który pozwala ukryć slajd w jednym z trybów prezentacyjnych. W powyższym przykładzie pierwsza strona nie ukaże się w trybie materiałów do rozdania, gdyż w argumentcie otoczenia `frame` podano argument `<handout:0>`.

Warto zatytułować każdy ze slajdów, nie zaś jedynie slajd tytułowy. Zauważmy, że polecenia `\section` i `\subsection` nie służą do tytułowania slajdów; służy do tego instrukcja `\frametitle`. Gdyby potrzebny był podtytuł, to – jak pokazano w przykładzie – można użyć otoczenia `block`.

Użycie polecenia `\pause` w otoczeniu `itemize` pozwala rozwijać punkty jeden po drugim, w miarę postępów prezentacji. Dodatkowe efekty prezentacyjne można osiągnąć za pomocą instrukcji: `\only`, `\uncover`, `\alt` oraz `\temporal`. Do dalszego sterowania prezentacją służą dopuszczalne w wielu miejscach opcje, podawane w nawiasach kątowych.

Cokolwiek mówić, aby uzyskać pełny obraz wszystkich dostępnych parametrów, trzeba przeczytać dokumentację `beameruserguide.pdf` klasy `beamer`. Pakiet ten jest ciągle rozwijany, dlatego warto po nowości zajrzeć na stronę <http://latex-beamer.sourceforge.net/>.

4.9. Pakiet pdfscreen

Pakiet `pdfscreen`, opracowany przez C.V. Radhakrishnana, pozwala tworzyć dokumenty PDF „zorientowane ekranowo”, to znaczy takie, które będzie się wygodnie czytało z ekranu monitora (inne wymiary kolumny tekstu, większy krój pisma). Takie dokumenty mogą też zawierać różne elementy nawigacyjne przeznaczone do poruszania się po dokumencie. Efektowne przykłady dokumentów przygotowanych za pomocą pakietu `pdfscreen` można znaleźć pod adresem <http://www.tug.org.in/tutorial/>. Szczegółowy polski opis pakietu zawiera [24].

```
\begin{tabular}{c r @{\,} l}
Wyrażenie & & Wartość \\
\multicolumn{2}{c}{\hrline}
$\pi$ & 3&1416 & \\
$\pi^{\pi}$ & 36&46 & \\
$\pi^{\pi^{\pi}}$ & 80662&7 & \\
\end{tabular}
```

Wyrażenie	Wartość
π	3,1416
π^π	36,46
$(\pi^\pi)^\pi$	80662,7

Polecenie `\cline{m-n}` wstawia poziomą kreskę ciągnącą się od kolumny m do kolumny n :

```
\begin{tabular}{|c|c|c|c|l|}\hrline
1 & \multicolumn{4}{c}{0} \\
\cline{2-5}
1 & 2 & 3 & 4 & 5 \\
\cline{2-4}
1 & 2 & 3 & 4 & 5 \\
\hrline
\end{tabular}
```

1	0			
1	2	3	4	5
1	2	3	4	5

W punkcie 6.8 opisano, jak sobie radzić ze składem bardziej skomplikowanych tabel.

2.12. Wstawki

Współczesne publikacje zawierają dużo rysunków i tabel. Elementów tych nie należy dzielić między strony i dlatego wymagają specjalnego potraktowania. W sytuacji gdy nie mieszczą się one na bieżącej stronie, na ogół przenosi się je i wstawia na początku strony następnej. W wypadku przeniesienia rysunku lub tabeli miejsce pozostałe na stronie wypełniane jest tekstem. Tego typu elementy będziemy nazywać *wstawkami*.

Aby w pełni skorzystać z \LaTeX owego mechanizmu wstawek, trzeba choćby powierzchownie rozumieć, jak \LaTeX manipuluje takimi obiektami. W przeciwnym razie wstawki mogą się stać źródłem irytacji, gdyż \LaTeX będzie je umieszczał wszędzie, tylko nie w miejscach, w którym byśmy sobie tego życzyli.

Do tworzenia wstawek mamy w \LaTeX u dwa otoczenia. Otoczenie `figure` służy do tworzenia rysunków, a otoczenie `table` – do tabel. Oba mają jeden parametr opcjonalny:

```
\begin{figure}[miejsce] albo \begin{table}[miejsce]
```

Argument *miejsce* określa, gdzie na stronie można umieścić wstawkę. Powinna to być sekwencja od jednego do pięciu znaków: `h`, `t`, `b`, `p` oraz `!`. Każdy znak określa dopuszczalny sposób umieszczenia wstawki; szczegółowe informacje na ten temat zestawiono w tabeli 2.3.

Przykładowa tabela może się zaczynać tak:

```
\begin{table}![hbp]
```

od ustawień podanych wraz z klasą dokumentu. Kod podany powyżej pozwala użyć zarówno instrukcji `pdflatex` do wygenerowania PDF-a, jak i `latex` do uzyskania pliku DVI.

4.7.8. Wymiary kartki papieru

W \LaTeX u rozmiar papieru zadaje się w opcjonalnym argumencie polecenia `\documentclass`, na przykład `a4paper` albo `letterpaper`. Opcje te działają również w wypadku generowania dokumentu PDF, tyle że ich wykonanie nie wystarcza do stworzenia poprawnego pliku PDF. Jeśli korzystamy z pakietu `hyperref`, to niezbędne polecenia zostaną wykonane automatycznie. W przeciwnym razie musimy to zrobić ręcznie, umieszczając w preambule dokumentu następujący fragment:

```
\ifpdf %% działa tylko z pdftex:
\pdfpagewidth=\paperwidth \pdfpageheight=\paperheight
%% działa z programami latex+dvips:
\else \special{papersize=\paperwidth,\pageheight} \fi
```

4.8. Tworzenie prezentacji

Daniel Flipo <Daniel.Flipo@univ-lille1.fr>

Wyniki naszej pracy naukowej możemy przedstawiać kredą na tablicy, za pomocą rzutnika i przezroczcy (slajdów) bądź – posługując się odpowiednim oprogramowaniem – bezpośrednio z laptopa.

Program `pdf \LaTeX` w połączeniu z klasą `beamer` służy do tworzenia prezentacji w formacie PDF. Wyglądają one tak, jak gdybyśmy je wygenerowali – mając dobry dzień i sporo szczęścia – za pomocą PowerPointa, ale są bardziej przenośne, bo Acrobat Reader jest dostępny w większej liczbie systemów.

Klasa `beamer` używa pakietów `graphicx`, `color` oraz `hyperref` z opcjami zaadaptowanymi do prezentacji ekranowych.

Kompilując `pdf \LaTeX` em kod z rysunku 4.2, otrzymamy plik PDF ze stroną tytułową oraz jeszcze jedną stroną, zawierającą kilka punktów, które mają się odsłaniać w miarę, jak podczas prezentacji będziemy przechodzili do kolejnych jej kroków.

Jedną z zalet klasy `beamer` jest to, że generuje ona gotowy do użycia plik PDF, bez konieczności przechodzenia przez fazę postscriptową, jak to jest w wypadku pakietu `prosper` albo wymagającego dodatkowego przetworzenia pakietu `ppower4`. Korzystając z klasy `beamer`, możemy z tego samego pliku źródłowego generować kilka wersji prezentacji, tak zwanych trybów. Plik źródłowy może w nawiasach kątowych zawierać instrukcje przeznaczone do różnych trybów. Dostępne są następujące tryby: `beamer` – dla omówionych wyżej prezentacji PDF, `trans` – do slajdów, oraz `handout` – do wydruku. Domyślnym jest `beamer`, a inny tryb możemy zadać jako opcję

Następujące instrukcje:

```
\listoffigures oraz \listoftables
```

działają analogicznie do instrukcji `\tableofcontents`, wstawiając do dokumentu, odpowiednio, spis rysunków oraz spis tabel. Poszczególnymi pozycjami tych spisów będą tytuły rysunków bądź tabel będące argumentami instrukcji `\caption`. Jeżeli tytuł jest długi, to do spisu można przesłać jego wersję skróconą, podaną jako opcjonalny argument instrukcji `\caption`:

```
\caption[Short]{LLLLLoooooonnnnnnggggg}
```

Za pomocą instrukcji `\label` oraz `\ref` można tworzyć odsyłacze do tabel i rysunków.

Polecenie `\label` należy umieszczać *bezpośrednio* za instrukcją `\caption`. Dobrym pomysłem jest też umieszczenie jej wewnątrz argumentu instrukcji `\caption` (na przykład na końcu tytułu rysunku czy tabeli). Niektórzy użytkownicy błędnie sądzą, że wystarczy umieścić instrukcję `\label` wewnątrz otoczenia `figure` czy `table`, gdy tymczasem umieszczenie jej przed poleceniem `\caption` prowadzi do błędów w numerach odsyłaczy.

W poniższym przykładzie wstawka zawiera prostokąt o wymiarach 5 cm × 5 cm. Ten sposób postępowania można wykorzystać w celu zarezerwowania miejsca na rysunki, które zostaną wklejone później – do gotowego, wydrukowanego dokumentu.

Rysunek `\ref{white}` jest przykładem Pop-Artu.

```
\begin{figure}[!htp]
\makebox[\textwidth]{\framebox[5cm]{\rule{Opt}{5cm}}}
\caption{Pięć na pięć centymetrów\label{white} }
\end{figure}
```

Zakładając w tym przykładzie, że kolejka rysunków jest pusta, \LaTeX najpierw spróbuje umieścić rysunek bez przesuwania go dokądkolwiek (**h**). Jeżeli okaże się to niemożliwe, to spróbuje go umieścić na górze strony (**t**). Jeżeli i to okaże się niewykonalne, to będzie się starał umieścić rysunek na stronie zawierającej wyłącznie wstawki (**p**). Jeżeli w kolejkach rysunków i tabel nie ma wstawek pozwalających wypełnić stronę, to \LaTeX rozpocznie nową stronę i spróbuje umieścić na niej rysunek, traktując go znowu tak, jakby właśnie pojawił się w tekście.

Czasami może wystąpić konieczność wykonania instrukcji:

```
\clearpage albo nawet \cleardoublepage
```

W wyniku jej zadziałania \LaTeX umieszcza w dokumencie wszystkie oczekujące w kolejkach wstawki, a następnie rozpoczyna skład od nowej strony. W wypadku użycia instrukcji `\cleardoublepage` \LaTeX rozpoczyna skład od strony nieparzystej (por. przypis 2 na str. 18).

W dalszej części książki przedstawimy, jak można do dokumentu dołączyć rysunki w formacie Postscript (por. 4.1).

```
\author{Maria Oetiker %<\href{mailto:mary@oetiker.ch}%
{mary@oetiker.ch}>}%
```

Zauważmy, że łącze zapisano tu tak, iż adres elektroniczny pojawia się zarówno w łańcuchu, jak i na stronie. Gdyby podać go tak:

```
\href{mailto:mary@oetiker.ch}{Maria Oetiker}
```

to łącze działałoby w programie Acrobat, lecz adres nie byłby widoczny w wydruku.

4.7.5. Problemy z łączami

Ukazaniu się komunikatu w rodzaju:

```
! pdfTeX warning (ext4): destination with the same
  identifier (name{page.1}) has been already used,
  duplicate ignored
```

towarzyszy reinicjalizacja licznika, na przykład w wyniku użycia polecenia `\mainmatter`, dostępnego w klasie dokumentów `book`. Ustala ono wartość licznika stron na 1 tuż przed pierwszym rozdziałem książki. Ale ponieważ również i wstęp do książki zawiera stronę 1, to odsyłacz „page.1” staje się niejednoznaczny, stąd notka „duplicate ignored”.

Możemy temu przeciwdziałać, dodając opcję `plainpages=false`. Pomaga to niestety tylko w odniesieniu do licznika stron. Radykalniejszym rozwiązaniem jest opcja `hypertexnames=false`, która jednak powoduje, że przestają działać odsyłacze do stron w skorowidzu.

4.7.6. Problemy z zakładkami

Tekst na zakładkach nie zawsze wygląda zgodnie z naszymi oczekiwaniami. Ponieważ zakładki są traktowane jako „czysto tekstowe”, może w nich wystąpić mniejszy zakres znaków niż w normalnym tekście \LaTeX owym. Pakiet `hyperref` zauważy tego typu problem i zasygnalizuje go komunikatem:

```
Package hyperref Warning:
Token not allowed in a PDFDocEncoded string:
```

Możemy ten problem obejść, podając napis przeznaczony na zakładkę, który ma zastąpić wywołujący trudność tekst:

```
\texorpdfstring{ $T_{\mathbb{E}}$  tekst}{Napis na zakładkę}
```

Podstawowym kandydatem do zastosowania takiego postępowania są wzory matematyczne:

```
\section{\texorpdfstring{\mathbb{E}^2}{E=mc^2}}
```

Śródtytuł „ $E=mc^2$ ” przekształca się na zakładce w „E=mc2”.

Także zmiany kolorów nie przenoszą się dobrze na zakładki:

```
\section{\textcolor{czerwony}{Czerwony !}}
```

Deklaracje `\address`, `\signature` oraz `\date` umieszcza się zwykle w preambule, chociaż mogą one wystąpić również w części zasadniczej pliku źródłowego.

Treść listu powinniśmy wpisać wewnątrz otoczenia `letter`. Otoczenie to ma jeden argument, którym jest adres osoby, do której piszemy. Wewnątrz otoczenia `letter` można stosować kilka prostych poleceń służących do umieszczania w odpowiednim miejscu elementów typowego listu¹³. Do złożenia nagłówka listu używamy polecenia `\opening`, a do zakończenia – `\closing`. Ponadto są polecenia: `\ps` do wstawienia *post scriptum* oraz `\cc` do zdefiniowania wykazu osób, które mają otrzymać kopię listu. Oto pełny przykład listu:

```
%& --translate-file=il2-pl
\documentclass{letter}
\usepackage{polski}
\address{Dyr. E.~K.~Tor,\\ Przewodniczący Zastępcy\\
  Firma z~o.o.\\ w/m}
\signature{E.~K.~Thor}
\begin{document}
% pierwszy list
\begin{letter}{Henryk Potrykus\\ul.~Krótka\\Puck}
\opening{Szanowny Panie}
Z~przykrością zawiadamiam, że Pańskie podanie
zostało...
\closing{Z~poważaniem}
\cc{cc: Józef Wujke}
\end{letter}

% drugi list
\begin{letter}{Zofia Potrykus\\ul.~Szkolna\\Reda}
\opening{Szanowna Pani}
Odpowiadając na Pani pismo...
...
\end{letter}
\end{document}
```

¹³Elementy nietypowe zawsze można umieścić, korzystając z innych poleceń poznanych w tym rozdziale.

Prostym sposobem wyjścia z tej sytuacji jest konwersja plików EPS na format PDF programem `epstopdf`, dostępnym na wielu platformach. Dla grafiki wektorowej (rysunków) jest to wspaniałe rozwiązanie. Dla mapek bitowych (zdjęcia, skany) nie jest ono idealne, bo format PDF z natury obsługuje włączanie obrazów w formatach PNG i JPEG. Format PNG jest dobry do zdjęć z ekranów i innych obrazów z niewielką liczbą kolorów. Format JPEG jest świetny do zdjęć, jako że zużywa na nie mało pamięci.

Niekiedy zamiast rysowania figur geometrycznych korzystniejsze jest opisywanie ich w wyspecjalizowanym języku poleceń, na przykład takim, jaki jest stosowany w programie `METAPOST`, który – wraz z obszernym podręcznikiem – wchodzi w skład większości dystrybucji `TEX`.

4.7.4. Łącza hipertekstowe

O kierowanie wewnętrznych odsyłaczy we właściwe miejsca dokumentu za-dba pakiet `hyperref`. Aby zadziałał poprawnie, zaleca się umieścić polecenie `\usepackage{hyperref}` jako *ostatnie* w preambule dokumentu. Sposobem działania pakietu `hyperref` można sterować za pomocą wielu opcji, które po-dajemy bądź „tradycyjne”, czyli jako listę rozdzielonych przecinkami opcji po `\usepackage`, wewnątrz nawiasów kwadratowych, bądź jako argument polecenia `\hypersetup{opcje}`.

Opcje `pdftex` i `dvips` są kluczowe, bo określają metodę generowania dokumentu PDF: bezpośrednio za pomocą programu `pdfTEX` albo pośrednio poprzez zamianę wynikowego pliku DVI na PS, a potem PDF.

W poniższym wykazie wartości domyślne są podane pismem prostym.

`bookmarks (=true, false)` w trakcie wyświetlania dokumentu Acrobatem pokaż bądź ukryj pasek zakładek;
`unicode (=false, true)` pozwól w zakładkach Acrobatu używać znaków z alfabetów nielacińskich;
`pdftoolbar (=true, false)` pokaż bądź ukryj pasek narzędziowy Acrobatu;
`pdfmenubar (=true, false)` pokaż bądź ukryj menu Acrobatu;
`pdffitwindow (=true, false)` dostosuj wielkość wyświetlanego PDF-a do wielkości okna;
`pdftitle (= {napis})` tytuł dokumentu;
`pdfauthor (= {napis})` nazwisko autora;
`pdfnewwindow (=true, false)` określa, czy w wypadku gdy łącze prowadzi poza dokument, ma być otwierane nowe okna;
`colorlinks (=false, true)` określa, czy otoczyć hiperłącza kolorowymi ramkami (`false`) czy kolorować same hiperłącza (`true`). Kolory można konfigurować za pomocą następujących opcji (w nawiasach kolory domyślne):
`linkcolor (=red)` kolor łączy wewnętrznych (rozdziałów, punktów, stron itp.),

```
$a$ do kwadratu plus $b$
do kwadratu równa się $c$
do kwadratu. Albo,
bardziej matematycznie:
\begin{displaymath}
c^2=a^2+b^2
\end{displaymath}
Pierwszy wiersz po wzorze.
```

a do kwadratu plus b do kwadratu równa się c do kwadratu. Albo, bardziej matematycznie:

$$c^2 = a^2 + b^2$$

Pierwszy wiersz po wzorze.

Do uzyskiwania wzorów numerowanych stosujemy otoczenie `equation`. Instrukcji `\label` możemy wówczas użyć do zapamiętania numeru wzoru, a polecenia `\ref` albo pochodzącego z pakietu `amsmath` `\eqref` – do przywołania w dokumencie tego numeru:

```
\begin{equation}
\epsilon > 0 \label{eq:eps}
\end{equation}
Ze wzoru (\ref{eq:eps})
otrzymujemy \ldots
```

$\epsilon > 0$ (3.1)

Ze wzoru (3.1) otrzymujemy ...

Zwróćmy uwagę na różnicę w wyglądzie wzorów złożonych wewnątrz akapitu i w wersji eksponowanej:

```
$$\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}$$
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\lim_{n \to \infty}
\sum_{k=1}^n \frac{1}{k^2}
= \frac{\pi^2}{6}
\end{displaymath}
```

$$\lim_{n \rightarrow \infty} \sum_{k=1}^n \frac{1}{k^2} = \frac{\pi^2}{6}$$

Między *trybem matematycznym* \LaTeX a a *trybem tekstowym* istnieją znaczne różnice. Na przykład, w *trybie matematycznym*:

1. \LaTeX ignoruje prawie wszystkie odstępy oraz znaki końca linii; wszystkie odstępy we wzorach wynikają bądź z kontekstu, bądź z użycia specjalnych poleceń, takich jak: `\`, lub `\quad` (por. punkt 6.3.3, str. 110).
2. Puste linie są niedozwolone. Nie ma czegoś takiego, jak podział wzorów na akapity.
3. Litery we wzorach służą do oznaczania nazw zmiennych; zmienne składamy inaczej niż zwykły tekst. Jeżeli częścią wzoru ma być zwykły tekst, to należy się posłużyć instrukcją `\text{trm}{...}`.

```
\begin{equation}
\forall x \in \mathbf{R} \colon
\quad x^2 \geq 0
\end{equation}
```

$\forall x \in \mathbf{R}: \quad x^2 \geq 0$ (3.2)

dokumentu PDF za pomocą programu pdf \TeX . Pierwszy z wymienionych sposobów, mimo że na pierwszy rzut oka wydaje się bardziej pracochłonny, ma tę przewagę nad drugim, że umożliwia wykorzystanie wielu wartościowych pakietów stosujących język Postscript, choćby takich jak `rotate` czy `pstricks`. Z kolei utworzenie pliku PDF bezpośrednio z pliku źródłowego, za pomocą programu pdf \TeX , jest prostsze. Którąkolwiek z metod wybierzesz, w większości przypadków efekt końcowy będzie identyczny.

Program pdf \TeX , opracowany przez Hàn Thê Thànha, generuje dokument PDF na podobnej zasadzie, jak \TeX produkuje plik DVI. Istnieje też pdf \LaTeX , który tworzy pliki PDF ze źródeł \LaTeX owych. Zarówno pdf \TeX , jak i pdf \LaTeX wchodzi w skład wszystkich współczesnych dystrybucji \TeX a, takich jak: te \TeX , fp \TeX , Mik \TeX , \TeX Live czy CMac \TeX .

Do wygenerowania pliku PDF zamiast DVI wystarczy zastąpić polecenie `latex file.tex` instrukcją `pdflatex file.tex`. W zintegrowanym środowisku graficznym, takim jak TeXnicCenter czy Kile, uruchomienie obu programów polega na wybraniu odpowiedniego przycisku z menu.

Tworzenie dobrej jakości dokumentów PDF wymaga: użycia *odpowiednich* fontów, dołączenia grafiki we *właściwym* formacie oraz zdefiniowania hiperłączy dla elementów takich jak odsyłacze, spisy czy skorowidze. W kolejnych punktach omówimy te zagadnienia szczegółowo. Jeszcze bardziej detaliczny opis można znaleźć w [24].

4.7.2. Fonty bitmapowe i obwiedniowe

Fonty są bardzo ważną częścią każdego systemu składu. Font to w istocie program komputerowy, zawierający opisy znaków oraz dodatkowe informacje określające sposób ich drukowania i pozycjonowania na stronie. Większość fontów jest komercyjna i nie może być swobodnie rozpowszechniana. Z tego powodu system \TeX nie używa fontów systemowych, na przykład fontów dostępnych w systemie MS Windows, lecz wykorzystuje alternatywne kroje pisma o statusie Oprogramowania Otwartego. Inaczej, dokumentu przygotowanego na przykład w systemie MS Windows nie można by poprawnie przetworzyć w systemie Linux albo, jeżeli nawet dałoby się go przetworzyć, to otrzymalibyśmy dokument wyglądający inaczej⁸.

Z punktu widzenia „technologii komputerowej” fonty dzielimy na bitmapowe i obwiedniowe, przy czym te pierwsze – za wyjątkiem systemu \TeX – mają znaczenie historyczne. Istnieje kilka standardów tworzenia fontów obwiedniowych: fonty postscriptowe (tzw. fonty Type 1 albo *typu pierwszego*), fonty TrueType oraz najnowsze OpenType. „Klasyczne” bitmapowe fonty \TeX a określa się akronimem PK.

⁸Inne na przykład byłoby łamanie wierszy i podział na strony. W edytorze MS Word nie jest to zresztą uważane za błąd, bo w tamtym programie dokument jest przenośny, ale tylko razem z komputerem, bo przesłanie dokumentu na inny komputer z reguły powoduje, że jest on złożony inaczej.

```
$_{1} x^{2} e^{-\alpha t}$  
$a^{3}_{ij} e^{x^2} \neq e^{x^2}$
```

$$a_1 x^2 e^{-\alpha t} a_{ij}^3 e^{x^2} \neq e^{x^2}$$

Pierwiastek kwadratowy składamy poleceniem `\sqrt`. Wielkość znaku pierwiastka jest przez \LaTeX a ustalana automatycznie. Zapis samego znaku pierwiastka umożliwia instrukcja `\surd`³, natomiast pierwiastek stopnia n składamy konstrukcją `\sqrt[n]`:

```
$_{\sqrt{x}} \sqrt{x^2 + \sqrt{y}}$  
$_{\sqrt[3]{2}} \surd[x^2 + y^2]$
```

$$\sqrt{x} \sqrt{x^2 + \sqrt{y}} \sqrt[3]{2} \sqrt{x^2 + y^2}$$

Polecenia `\overline` oraz `\underline` umieszczają nad i pod wyrażeniami poziome kreski:

```
$_{\overline{m+n}} \underline{x+y}$
```

$$\overline{m+n} \underline{x+y}$$

Instrukcje `\overbrace` oraz `\underbrace` umieszczają nad i pod wyrażeniami poziome klamry:

```
$_{\underbrace{a+b+\cdots+z}_{26}}$
```

$$\underbrace{a+b+\cdots+z}_{26}$$

Akcenty matematyczne, takie jak daszki czy tyldy nad zmiennymi, umieszczamy we wzorze poleceniami z tabeli 3.1. Szerokie daszki i tyldy, obejmujące wiele symboli, wstawiamy za pomocą instrukcji `\widetilde` oraz `\widehat`. Znakiem ' oznaczamy symbol „prim”:

```
\begin{displaymath}$  
$\hat{y} = x^2 \quad y' = 2x''$  
\end{displaymath}$
```

$$\hat{y} = x^2 \quad y' = 2x''$$

Wektory oznacza się niekiedy akcentem w postaci strzałki nad nazwą zmiennej. Służy do tego polecenie `\vec`. Natomiast do oznaczenia wektora od punktu A do punktu B korzystamy z poleceń `\overrightarrow` oraz `\overleftarrow`:

```
\begin{displaymath}$  
$\vec{a} \quad \overrightarrow{AB}$  
\end{displaymath}$
```

$$\vec{a} \quad \overrightarrow{AB}$$

Nazwy funkcji typu „logarytm” należy składać odmianą prostą, nie zaś kursywą, zarezerwowaną dla nazw zmiennych. Oto lista poleceń \LaTeX a służących do składu rozmaitych funkcji matematycznych:

³Taki zapis jest wykorzystywany raczej w literaturze anglosaskiej.

o rozszerzeniu `.dtx`. Często jest też dołączany plik `readme.txt` z krótkim opisem pakietu. Ma się rozumieć, zawsze warto zaczynać od przeczytania tego właśnie pliku.

Po skopiowaniu plików pakietu na nasz komputer trzeba je w ten czy inny sposób przetworzyć, aby po pierwsze wprowadzić do dystrybucji \TeX a informację o nowym pakiecie, a po drugie uzyskać dokumentację. Oto, jak osiąga się pierwszy z tych celów:

1. Uruchamiamy \LaTeX a na pliku `.ins`. To powoduje wygenerowanie pliku `.sty`.
2. Plik `.sty` kopiujemy w miejsce, w którym nasza dystrybucja \TeX a potrafi go odnaleźć. Zazwyczaj jest to katalog o nazwie `.../localtexmf/tex/latex` (użytkownicy systemów MS Windows oraz OS/2 wiedzą, że w ich systemie trzeba używać znaku ukośnika pochylonego przeciwnie).
3. Odświeżamy zawartą w dystrybucji bazę danych nazw plików. Odpowiednie polecenie zależy od dystrybucji \TeX a: w $\text{te}\text{\TeX}$ u i $\text{fp}\text{\TeX}$ u jest to `maktexlsr`; a w $\text{Mik}\text{\TeX}$ u – `initexmf -update-fndb`. W dystrybucjach wyposażonych w środowisko graficzne (np. $\text{Mik}\text{\TeX}$) do odświeżenia bazy nazw wystarczy wybrać odpowiednią pozycję z menu.

Można teraz wygenerować dokumentację z pliku `.dtx`:

1. Uruchamiamy \LaTeX a na pliku `.dtx`, w wyniku czego powstanie plik `.dvi`. Niewykluczone, że zanim w dokumentacji uporządkują się numery odsyłaczy, będziemy musieli uruchomić \LaTeX a kilka razy.
2. Sprawdzamy, czy wśród wielu plików, które wygenerował \LaTeX , jest plik `.idx`. Jeśli go nie widać, to możemy przejść do kroku 5.
3. Aby wygenerować skorowidz, piszemy linijkę rozkazową:

```
makeindex -s gind.ist nazwa
```

gdzie *nazwa* oznacza nazwę głównego spośród przetwarzanych plików, bez rozszerzenia.

4. Ponownie uruchamiamy \LaTeX a na pliku `.dtx`.
5. Na koniec coś równie ważnego: aby uprzyjemnić sobie czytanie, tworzymy plik `.ps` albo `.pdf`.

Czasami wśród wygenerowanych plików znajdziemy jeszcze plik `.glo`. W takiej sytuacji między krokami 4 i 5 powinniśmy wykonać:

```
makeindex -s gglo.ist -o name.gls name.glo
```

Zawsze przed przejściem do kroku 5 musimy mieć pewność, że w poprzednim kroku uruchomiliśmy \LaTeX a na pliku `.dtx`.

Do uzyskiwania symboli relacji binarnych może się przydać instrukcja `\stackrel`. Składa ona swój pierwszy argument czcionką pomniejszoną, jaka stosowana jest do indeksów, i umieszcza go nad drugim argumentem, złożonym czcionką normalnej wielkości:

```
\begin{displaymath}
\int f_N(x) \stackrel{!}{=} 1
\end{displaymath}
```

$$\int f_N(x) \stackrel{!}{=} 1$$

Znak całki składamy poleceniem `\int`, **znak sumowania** instrukcją `\sum`, zaś **operator iloczynu** za pomocą instrukcji `\prod`. Górne granice całkowania i sumowania określamy za pomocą `^`, a dolne – znakiem `_`, czyli podobnie jak w wypadku indeksów górnych i dolnych⁴:

```
\begin{displaymath}
\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \prod_{\epsilon}
\end{displaymath}
```

$$\sum_{i=1}^n \int_0^{\frac{\pi}{2}} \prod_{\epsilon}$$

Pakiet `amsmath` zawiera dwa dodatkowe narzędzia do sterowania położeniem indeksów w złożonych wyrażeniach: instrukcję `\substack` i otoczenie `subarray`:

```
\begin{displaymath}
\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) =
\sum_{\begin{subarray}{l} i \in I \\ 1 < j < m \end{subarray}} Q(i, j)
\end{displaymath}
```

$$\sum_{\substack{0 < i < n \\ 1 < j < m}} P(i, j) = \sum_{\substack{i \in I \\ 1 < j < m}} Q(i, j)$$

Do składu **nawiasów** i innych **ograniczników** typu ([< || ↑ mamy różnorodność symboli. Nawiasy okrągłe i kwadratowe wstawiamy bezpośrednio z klawiatury. Do nawiasów klamrowych stosujemy `\{` oraz `\}`. Wszystkie inne ograniczniki wstawiamy, używając specjalnych poleceń, np. `\updownarrow`. Zestawienie dostępnych ograniczników znajduje się w tabeli 3.7 na stronie 61.

```
\begin{displaymath}
\{a, b, c\} \neq \{a, b, c\}
\end{displaymath}
```

$$a, b, c \neq \{a, b, c\}$$

Poprzedzenie otwierającego ogranicznika poleceniem `\left`, a zamykającego poleceniem `\right` powoduje automatyczne ustalenie jego rozmiaru w zależności od wielkości zawartego między nimi wyrażenia. Uwaga: każde użycie

⁴`AMS-LATEX` ma dodatkowo wielolinijkowe indeksy dolne i górne.

Potrafi on tworzyć skorowidz według reguł angielskich jak i polskich. Program ten jest dostępny na przykład pod adresem <http://www.ia.pw.edu.pl/~wujek/tex/idx/plmindex.zip>. Więcej informacji na temat polskich zasad tworzenia skorowidzów można znaleźć w [17].

Umieszczając w dokumencie polecenie `\index`, powinniśmy zwrócić uwagę na odstęp. Oto przykład, jak może to wpłynąć na skład:

```
Moje słowo \index{słowo}. Inaczej
niż słowo\index{słowo}. Zwróćmy
uwagę na pozycję kropki.
```

```
Moje słowo . Inaczej niż słowo. Zwróćmy
uwagę na pozycję kropki.
```

4.4. Paginy górne i dolne

Pakiet `fancyhdr` (autor Piet van Oostrum), który można znaleźć w katalogu: `CTAN://macros/latex/contrib/fancyhdr/`, udostępnia polecenia do definiowania zawartości pagin. Zwróćmy uwagę na różnice w wyglądzie pagin na stronach: bieżącej i poprzedniej. Oprócz numeru strony, w paginie górnej na stronie parzystej znajduje się tytuł rozdziału, a na stronie nieparzystej dodatkowo tytuł punktu. Pageń, której zawartość podąża za treścią dokumentu, nazywa się fachowo żywą pageńką.

```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% zmiana liter w żywej pageńce na małe
\renewcommand{\chaptermark}[1]{\markboth{#1}{}}
\renewcommand{\sectionmark}[1]{\markright{\thesection\ #1}}
\fancyhf{} % usuń bieżące ustawienia pagin
\fancyhead[LE,RO]{\small\bfseries\thepage}
\fancyhead[LO]{\small\bfseries\rightmark}
\fancyhead[RE]{\small\bfseries\leftmark}
\renewcommand{\headrulewidth}{0.5pt}
\renewcommand{\footrulewidth}{0pt}
\addtolength{\headheight}{0.5pt} % pionowy odstęp na kreskę
\fancypagestyle{plain}{%
  \fancyhead{} % usuń p. górne na stronach pozbawionych
  % numeracji (plain)
  \renewcommand{\headrulewidth}{0pt} % pozioma kreska
}
```

Rysunek 4.1: Przykład wykorzystania pakietu `fancyhdr`

do utworzenia zwykłego odstęp międzywyrazowego; `\quad` – odstęp równego 1 em (—), a `\qquad` – dwóm em (— —). Instrukcja `\!` wstawia odstęp „ujemny”, to znaczy zamiast zwiększać, zmniejsza odstęp między znakami. Wielkość tego odstęp wynosi $-\frac{3}{18}$ em (—):

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\int\!\!\!\int_D g(x,y)
\, \, \ud x \, \, \ud y
\end{displaymath}
%
zamiast
\begin{displaymath}
\int\int_D g(x,y)\ud x \ud y
\end{displaymath}
```

zamiast

$$\iint_D g(x,y) dx dy$$

$$\int \int_D g(x,y) dx dy$$

Zwróćmy uwagę, że litera „d” w symbolu różniczki jest złożoną odmianą prostą pisma⁷.

Dzięki zdefiniowanym w pakiecie $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ ⁸ takim instrukcjom jak: `\iint`, `\iiint`, `\iiiiiint` oraz `\idotsint` powyższy przykład można złożyć dużo prościej:

```
\newcommand{\ud}{\mathrm{d}}
\begin{displaymath}
\iint_D \, \, \ud x \, \, \ud y
\end{displaymath}
```

$$\iint_D dx dy$$

Więcej wiadomości na ten temat znajdziemy w pliku `testmath.tex`, który jest częścią pakietu $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ lub w rozdziale ósmym podręcznika [6].

3.5. Wyrównywanie w pionie

Do składania macierzy używa się otoczenia `array`. Działa ono podobnie do wcześniej omówionego otoczenia `tabular`. Używane w przykładzie polecenie `\\` oznacza przejście do nowego wiersza macierzy:

```
\begin{displaymath}
\mathbf{X} =
\left( \begin{array}{ccc}
x_{11} & x_{12} & \dots \\
x_{21} & x_{22} & \dots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \dots \\ x_{21} & x_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

⁷W Polsce do składu litery „d” w różniczkach stosuje się kursywę matematyczną.

⁸Ścisłej mówiąc, w pakiecie `amsmath`.

Partl[~]`\cite{pa}` zaproponował, żeby `\ldots`

Partl [1] zaproponował, żeby ...

```
\begin{thebibliography}{99}
\bibitem{pa} H.~Partl:
\emph{German \TeX},
TUGboat Vol.~9, No.~1 ('88)
\end{thebibliography}
```

Bibliografia

[1] H. Partl: *German T_EX*, TUGboat Vol. 9, No. 1 ('88)

Do większych projektów przydaje się program o nazwie BibT_EX. Program ten znajduje się w każdej współczesnej dystrybucji T_EXa. BibT_EX korzysta z bazy bibliograficznej (biblioteki), z której wybiera tylko te pozycje literaturowe, które były cytowane w dokumencie. Sposób formatowania spisów literatury jest sterowany za pomocą specjalnych szablonów, których modyfikacja umożliwia zmianę układu graficznego spisu.

Po przetworzeniu pliku, na podstawie zawartości etykiet zapisanych przez L^AT_EXa do pliku `.aux`, BibT_EX tworzy spis literatury obejmujący tylko te pozycje z biblioteki, które cytowano w dokumencie. Zwykle plik ten ma rozszerzenie `.bib`. Format spisu zależy od specyfikacji szablonu znajdującej się w pliku o rozszerzeniu `.bst`, a jest zapisywany do pliku o rozszerzeniu `.bbl`. Do poprawnego sformatowania bibliografii i cytowań konieczne jest przynajmniej trzykrotne przetworzenie dokumentu L^AT_EXem. Więcej informacji na temat przygotowywania spisu literatury za pomocą BibT_EXa zawiera [1].

4.3. Skorowidze

Niezwykle użytecznym elementem wielu książek jest skorowidz. Można go utworzyć stosunkowo łatwo za pomocą L^AT_EXa oraz programu narzędziowego o nazwie `makeindex`⁶. W tym wprowadzeniu omówimy jedynie podstawowe polecenia dotyczące skorowidzów. Jak zawsze, więcej informacji znajdziemy w [6].

Generować hasła do skorowidza można dopiero po załadowaniu w preambule dokumentu pakietu o nazwie `makeidx`:

```
\usepackage{makeidx}
```

oraz wstawieniu (także w obrębie preambuły) instrukcji:

```
\makeindex
```

⁶Albo `makeidx`, jeśli nasz system operacyjny nie pozwala używać nazw dłuższych niż 8 znaków.


```
{\setlength\arraycolsep{2pt}
\begin{eqnarray}
\sin x & = & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\
& + \frac{x^7}{7!} - \dots
\end{eqnarray}}
```

$$\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots \quad (3.8)$$

```
\begin{eqnarray}
\left. \cos x = 1 - \frac{x^2}{2!} + \right. \\
\left. + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \right.
\end{eqnarray}}
```

$$\cos x = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots \quad (3.9)$$

\LaTeX nie generuje numeru w tym wierszu wyrażenia, w którym pojawia się polecenie `\nonumber`.

Złożenie w ten sposób dużych i skomplikowanych wyrażeń może się jednak okazać dość trudne. Lepiej wtedy użyć pakietu `amsmath`, w którym mamy do dyspozycji otoczenia `align`, `flalign`, `gather`, `multline` i `split`.

3.6. Fantomy

Fantomów nie można zobaczyć, mimo to zajmują one w naszych umysłach trochę miejsca. Nie inaczej jest w \LaTeX u, co pozwala robić z odstępami różne sztuczki.

Podczas wyrównywania w pionie tekstu z indeksami \sim bądź $_$ \LaTeX bywa nadgorliwy. Polecenie `\phantom` pozwala rezerwować miejsce na znaki, które nie mają się pojawić w ostatecznym wydruku. Najłatwiej to zrozumieć, analizując przykład:

```
\begin{displaymath}
{}^{12}_{6}\text{\phantom{C}}\text{\texttrm{C}} \\
\quad \quad \quad \text{\texttrm{versus}} \quad \quad \quad \\
{}^{12}_{6}\text{\texttrm{C}}
\end{displaymath}}
```

$${}^{12}_{6}\text{C} \quad \text{versus} \quad {}^{12}_{6}\text{C}$$

```
\begin{displaymath}
\Gamma_{ij}^k \text{\phantom{versus}} \Gamma_{ij}^k \\
\quad \quad \quad \text{\texttrm{versus}} \quad \quad \quad \\
\Gamma_{ij}^k
\end{displaymath}}
```

$$\Gamma_{ij}^k \quad \text{versus} \quad \Gamma_{ij}^k$$

```
\begin{figure}
%\begin{center} zamiast \begin{center} użyj lepiej
\centering % bo \centering nie wstawia dodatkowego odstępu
\includegraphics[angle=90,width=0.5\textwidth]{sowauszata.eps}
\end{figure}
```

W powyższym przykładzie do dokumentu jest dołączany rysunek z pliku `sowauszata.eps`. Rysunek najpierw obracamy o 90° w kierunku przeciwnym do ruchu wskazówek zegara, a następnie przeskalowujemy tak, by nadać mu szerokość równą połowie szerokości szpalty. Skalowanie grafiki jest równomierne z uwagi na brak klucza `height`. Szerokość i wysokość rysunku możemy też określić w jednostkach bezwzględnych, takich jak punkty czy centymetry. W tabeli 6.5 zestawiono jednostki miar w \LaTeX u. Więcej informacji na temat powyższych zagadnień znajdziemy w [2] i [22].

Z powodów opisanych w punkcie 4.7.3, *zaleca się* podawać nazwę dołączanego pliku graficznego *bez rozszerzenia*, to znaczy lepiej zapisać powyższe polecenie `\includegraphics` następująco:

```
\includegraphics[angle=90,width=0.5\textwidth]{sowauszata}
```

W takiej sytuacji \LaTeX będzie szukał pliku `sowauszata` o rozszerzeniu adekwatnym do możliwości zadeklarowanego w poleceniu `\usepackage` sterownika. Przykładowo, jeżeli `graphicx` było wywołane z opcją `dvips`, to szukany będzie plik `sowauszata.eps`, jeżeli zaś użyjemy opcji `pdftex`, to będą szukane pliki o rozszerzeniach `.pdf`, `.jpg` i `.png`, a pierwszy znaleziony zostanie dołączony.

Rysunki najlepiej jest umieszczać w oddzielnym katalogu, będącym podkatalogiem tego, w którym jest dokument \LaTeX a. Bez względu na używany system operacyjny katalogi w ścieżce dostępu należy odzielać znakiem $/^5$:

```
\includegraphics[width=0.5\textwidth]{rys/sowy/sowauszata}
```

Czasami wewnątrz jednego otoczenia `figure` chcemy umieścić kilka plików graficznych. W tym celu wystarczy, że umieścimy je obok siebie i w miarę potrzeby odpowiednio przeskalujemy:

```
\begin{figure}
\centering
% http://pl.wikipedia.org/wiki/Ptaki_Polski
\includegraphics[width=.3\textwidth]{sowauszata}
\includegraphics[width=.3\textwidth]{puszczykmszarny}
\includegraphics[width=.3\textwidth]{bubobubo}%puchacz
\end{figure}
```

W powyższym przykładzie trzy rysunki zostaną umieszczone jeden obok drugiego. Ich łączna szerokość wyniesie $3 \times 0,3 = 0,9$ szerokości kolumny

⁵Wszystkie uwagi dotyczące nazw plików opisane w punkcie 1.8 dla polecenia `\include` dotyczą także polecenia `\includegraphics`.

3.8. Twierdzenia, definicje, itp.

W pracach matematycznych występuje potrzeba wyróżniania w składzie zapisu lematów, definicji, aksjomatów i tym podobnych elementów. Do zdefiniowania nowego typu elementu służy polecenie:

```
\newtheorem{nazwa}[nazwa']{tekst}[punkt]
```

Argument *nazwa* oznacza nazwę otoczenia, zaś *tekst* jest napisem, który zostanie wydrukowany; może to być „Twierdzenie”, „Definicja” itp. Argumenty w nawiasach kwadratowych są nieobowiązkowe. Za ich pomocą określamy sposób numerowania twierdzeń. Opcjonalny argument *nazwa'* to nazwa elementu uprzednio zdefiniowanego poleceniem `\newtheorem`. Jeśli ten argument podano, to otoczenia *nazwa* oraz *nazwa'* będą posiadały wspólną numerację. Argument *punkt* określa sposób numerowania twierdzeń: jeżeli umieścimy tam na przykład `chapter`, to elementy będą numerowane w obrębie rozdziałów. Domyślnie otoczenia definiowane za pomocą `\newtheorem` są numerowane w sposób ciągły w obrębie całego dokumentu.

Po umieszczeniu instrukcji `\newtheorem{nazwa}...` w preambule można otoczenie *nazwa* stosować w następujący sposób:

```
\begin{nazwa}[tekst]
Oto moje interesujące twierdzenie
\end{nazwa}
```

Instrukcja `\newtheoremstyle{style}` z pakietu `amsthm` pozwala określić sposób formatowania twierdzenia przez wybór spośród trzech predefiniowanych stylów: `definition` (wytluszczony tytuł, treść złożona pismem prostym), `plain` (wytluszczony tytuł, treść zapisana kursywą) oraz `remark` (tytuł zapisany kursywą, treść – pismem prostym).

Tyle teoria. Poniższe przykłady usuną, miejmy nadzieję, wszelkie wątpliwości i jednocześnie uświadomią, że działanie otoczenia `\newtheorem` niełatwo zrozumieć:

```
% definicje w~preambule
\newtheorem{twr}{Twierdzenie}
\newtheorem{lem}[twr]{Lemat}
% po \begin{document}
\begin{lem} Pierszy
lemat\dots\label{lem:1} \end{lem}
\begin{twr}[Dyzma]
Przyjmując w~lemacie~\ref{lem:1},
że  $\epsilon=0\dots$  \end{twr}
\begin{lem}Trzeci lemat\end{lem}
```

Lemat 1. *Pierszy lemat...*

Twierdzenie 2 (Dyzma). *Przyjmując w lemacie 1, że $\epsilon = 0...$*

Lemat 3. *Trzeci lemat*

Elementy Twierdzenie i Lemat używają tego samego licznika. Argument nieobowiązkowy (wewnątrz nawiasów kwadratowych) służy do umieszczenia komentarza, w postaci nazwiska twórcy itp.

Rozdział 4

Rysunki, skorowidze, generowanie plików PDF...

Kolej teraz na opis możliwości \LaTeX a przydatnych w pracy nad większymi dokumentami, takich jak: dołączania rysunków w dokumencie, tworzenie skorowidzów i spisów literatury. Bardziej szczegółowy opis tych i pokrewnych zagadnień można znaleźć w [12] oraz [6]. Pod koniec niniejszego rozdziału jest też mowa o tym, jak \LaTeX może generować pliki PDF.

4.1. Włączanie grafiki w formacie EPS

Najprościej przygotować ilustrację w wyspecjalizowanym programie graficznym w rodzaju `xfig`, `CorelDraw!`, `Freehand`, `gnuplot`, itp., a później włączyć gotowy rysunek do dokumentu. Chociaż można to zrobić na wiele sposobów, tutaj przedstawimy jedynie sposób na dołączanie grafiki w formacie EPS (*Encapsulated PostScript*), jako że jest to technika prosta i szeroko stosowana¹. Do pracy z grafiką w formacie EPS potrzebujemy albo drukarki wyposażonej w język Postscript, albo programu `ghostscript`, dostępnego na przykład pod adresem <http://www.cs.wisc.edu/~ghost/>. Program `ghostscript` oraz ułatwiające posługiwanie się nim graficzne nakładki, takie jak: `ghostview`, `gv` czy `gsview` są dostępne (pod wyżej wymienionym adresem) na wszystkie popularne platformy systemowe.

Wielu poleceń przydatnych do włączania rysunków dostarcza pakiet `graphicx` (autor: D. P. Carlisle), będący częścią zestawu o nazwie „`graphics`”². Włączenie grafiki do dokumentu za jego pomocą możemy przedstawić w następujący sposób:

¹Użytkownicy programów typu Office, którzy w tym momencie być może po raz pierwszy usłyszeli, że istnieje coś takiego jak język Postscript i jego wariant EPS, mogą być tym stwierdzeniem zdziwieni. Tak jednak jest w istocie: inne standardy obowiązują w biurze, a inne w przemyśle poligraficznym. Więcej na temat grafiki można znaleźć w [19].

²Zestaw „`graphics`” jest obowiązkową (ang. *required*) częścią każdej dystrybucji \LaTeX a, można go też znaleźć w katalogu CTAN://`macros/latex/required/graphics`.

sposobem uzyskać symbole półgrube nawet wtedy, gdy w systemie brak odpowiednich fontów.

```
\begin{displaymath} \mu, M \quad \quad \quad \mu, M \quad \quad \quad \mu, M \\ \boldsymbol{\mu}, \boldsymbol{M} \quad \quad \quad \boldsymbol{\mu}, \boldsymbol{M} \quad \quad \quad \boldsymbol{\mu}, \boldsymbol{M} \\ \pmb{\mu}, \pmb{M} \quad \quad \quad \pmb{\mu}, \pmb{M} \quad \quad \quad \pmb{\mu}, \pmb{M} \end{displaymath}
```

$$\mu, M \quad \mu, M \quad \mu, M$$

3.10. Zestawienie symboli matematycznych

W poniższych tabelach zestawiono wszystkie symbole standardowo dostępne w trybie matematycznym. Symbole w tabelach 3.11–3.15¹⁰ są dostępne, jeżeli mamy zainstalowane dodatkowe fonty matematyczne (AMS *math fonts*) i do dokumentu dołączymy pakiet `amssymb`. W razie braku fontów lub pakietu można je odnaleźć w CTAN://macros/latex/required/amslatex. Ponadto dużo bardziej kompletne zestawienie symboli matematycznych można znaleźć w CTAN://info/symbols/comprehensive.

Tabela 3.1: Akcenty matematyczne

\hat{a} <code>\hat{a}</code>	\check{a} <code>\check{a}</code>	\tilde{a} <code>\tilde{a}</code>	\acute{a} <code>\acute{a}</code>
\grave{a} <code>\grave{a}</code>	\dot{a} <code>\dot{a}</code>	\ddot{a} <code>\ddot{a}</code>	\breve{a} <code>\breve{a}</code>
\bar{a} <code>\bar{a}</code>	\vec{a} <code>\vec{a}</code>	\widehat{A} <code>\widehat{A}</code>	\widetilde{A} <code>\widetilde{A}</code>

Tabela 3.2: Litery alfabetu greckiego

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

W trybie matematycznym \LaTeX wstawia dodatkowy mały odstęp po przecinku i średniku, natomiast w wypadku dwukropka wstawia odstęp

¹⁰Tabele przygotowano na podstawie pliku `symbols.tex` (David Carlisle), gruntownie zmodyfikowanego zgodnie z sugestiami Josefa Tkadleca.

Tabela 3.15: Negacje symbolów relacji i strzałek (pakiet AMS)

\nless <code>\nless</code>	\ngtr <code>\ngtr</code>	\varsubsetneqq <code>\varsubsetneqq</code>
\lneq <code>\lneq</code>	\gneq <code>\gneq</code>	\varsupsetneqq <code>\varsupsetneqq</code>
\lneq <code>\lneq</code>	\ngeq <code>\ngeq</code>	\subsetneqq <code>\subsetneqq</code>
\lneqslant <code>\lneqslant</code>	\ngeqslant <code>\ngeqslant</code>	\supsetneqq <code>\supsetneqq</code>
\lneqq <code>\lneqq</code>	\gneqq <code>\gneqq</code>	\nmid <code>\nmid</code>
\lvertneqq <code>\lvertneqq</code>	\gvertneqq <code>\gvertneqq</code>	\nparallel <code>\nparallel</code>
\lneqq <code>\lneqq</code>	\gneqq <code>\gneqq</code>	\nshortmid <code>\nshortmid</code>
\lnsim <code>\lnsim</code>	\gnsim <code>\gnsim</code>	\nshortparallel <code>\nshortparallel</code>
\lnapprox <code>\lnapprox</code>	\gnapprox <code>\gnapprox</code>	\nsim <code>\nsim</code>
\nprec <code>\nprec</code>	\nsucc <code>\nsucc</code>	\ncong <code>\ncong</code>
\npreceq <code>\npreceq</code>	\nsucceq <code>\nsucceq</code>	\nvdash <code>\nvdash</code>
\nprecneqq <code>\nprecneqq</code>	\nsuccneqq <code>\nsuccneqq</code>	\nVDash <code>\nVDash</code>
\nprecnsim <code>\nprecnsim</code>	\nsuccnsim <code>\nsuccnsim</code>	\nVdash <code>\nVdash</code>
\nprecnapprox <code>\nprecnapprox</code>	\nsuccnapprox <code>\nsuccnapprox</code>	\ntriangleleft <code>\ntriangleleft</code>
\subsetneq <code>\subsetneq</code>	\supsetneq <code>\supsetneq</code>	\ntriangleright <code>\ntriangleright</code>
\varsubsetneq <code>\varsubsetneq</code>	\varsupsetneq <code>\varsupsetneq</code>	\ntrianglelefteq <code>\ntrianglelefteq</code>
\subsetneqq <code>\subsetneqq</code>	\supsetneqq <code>\supsetneqq</code>	\ntrianglerighteq <code>\ntrianglerighteq</code>
\nleftarrow <code>\nleftarrow</code>	\nrightarrow <code>\nrightarrow</code>	\nleftrightarrow <code>\nleftrightarrow</code>
\nLeftarrow <code>\nLeftarrow</code>	\nRightarrow <code>\nRightarrow</code>	\nLeftrightarrow <code>\nLeftrightarrow</code>

Tabela 3.16: Relacje dwuargumentowe (pakiet AMS)

\dotplus <code>\dotplus</code>	\centerdot <code>\centerdot</code>	\intercal <code>\intercal</code>
\ltimes <code>\ltimes</code>	\rtimes <code>\rtimes</code>	\divideontimes <code>\divideontimes</code>
\Cup <code>\Cup</code>	\veebar <code>\veebar</code>	\smallsetminus <code>\smallsetminus</code>
\Cap <code>\Cap</code>	\barwedge <code>\barwedge</code>	\doublebarwedge <code>\doublebarwedge</code>
\boxplus <code>\boxplus</code>	\boxminus <code>\boxminus</code>	\circleddash <code>\circleddash</code>
\boxtimes <code>\boxtimes</code>	\boxdot <code>\boxdot</code>	\circledcirc <code>\circledcirc</code>
\leftthreetimes <code>\leftthreetimes</code>	\curlyvee <code>\curlyvee</code>	\circledast <code>\circledast</code>
\rightthreetimes <code>\rightthreetimes</code>	\curlywedge <code>\curlywedge</code>	

Tabela 3.4: Symbole operacji dwuargumentowych

$+$	$-$	
\pm	\mp	\triangleleft
\cdot	\div	\triangleright
\times	\setminus	\star
\cup	\cap	\ast
\sqcup	\sqcap	\circ
\vee	\wedge	\bullet
\oplus	\ominus	\diamond
\odot	\oslash	\uplus
\otimes	\bigcirc	\amalg
\triangleup	\triangledown	\dagger
\triangleleft	\triangleright	\ddagger
\trianglelefteq	\trianglerighteq	\wr

Tabela 3.5: Symbole zmiennej wielkości

\sum	\bigcup	\bigvee	\bigoplus
\prod	\bigcap	\bigwedge	\bigotimes
\coprod	\bigsqcup		\bigodot
\int	\oint		\biguplus

Tabela 3.6: Strzałki

\leftarrow	\longleftarrow	\uparrow
\rightarrow	\longrightarrow	\downarrow
\leftrightarrow	\longleftrightarrow	\updownarrow
\Lleftarrow	\Llongleftarrow	\Uparrow
\Rrightarrow	\Rlongrightarrow	\Downarrow
\Leftrightarrow	\Llongleftrightarrow	\Updownarrow
\mapsto	\longmapsto	\nearrow
\hookrightarrow	\hookrightarrow	\searrow
\leftharpoonup	\rightharpoonup	\swarrow
\leftharpoondown	\rightharpoondown	\nwarrow
\rightleftharpoons	\iff	\leadsto

^a Dostępne po dołączeniu pakietu latexsym.

Tabela 3.7: Ograniczniki

$($	$)$	\uparrow	\Uparrow
$[$	\lbrack	\rangle	\Downarrow
$\{$	\lbrack	\langle	\Updownarrow
$\}$	\rbrack	\rfloor	\lvert
$\}$	\rbrack	\lfloor	\lceil
$/$	$/$	\backslash	

Tabela 3.8: Duże ograniczniki

\lg	\rg	\lmoustache	\rmoustache
\arrowvert	\Arrowvert	\bracevert	

Tabela 3.9: Różne symbole

\dots	\cdots	\vdots	\ddots
\hbar	\imath	\jmath	ℓ
\Re	\Im	\aleph	\wp
\forall	\exists	\mho	∂
\prime	\prime	\emptyset	∞
∇	\triangle	\Box	\Diamond
\bot	\top	\angle	\surd
\diamondsuit	\heartsuit	\clubsuit	\spadesuit
\neg	\flat	\natural	\sharp

^a Dostępne po dołączeniu pakietu latexsym.

Tabela 3.10: Symbole niematematyczne

\dagger	\S	\copyright	\ddagger	\P	\pounds
-----------	------	--------------	------------	------	-----------

Polecenia te są dostępne również w trybie tekstowym.

Tabela 3.11: Ograniczniki (pakiet AMS)

\ulcorner	\urcorner	\llcorner	\lrcorner
-------------	-------------	-------------	-------------

Tabela 3.12: Symbole Greckie i Hebrajskie (pakiet AMS)

\digamma	\varkappa	\beth	\daleth	\gimel
------------	-------------	---------	-----------	----------