Przedmowa

LaTeX [4] jest systemem składu umożliwiającym między innymi tworzenie dokumentów naukowych i technicznych o wysokiej jakości typograficznej. Oczywiście, oprócz tego LaTeX umożliwia przygotowywanie dowolnego rodzaju dokumentów, poczynając od listów, a kończąc na grubych książkach. Do formatowania dokumentów LaTeX wykorzystuje program TeX [3].

Niniejsze krótkie *Wprowadzenie* opisuje LaTeX-a w zakresie wystarczającym do większości zastosowań. Pełny opis LaTeX-a można znaleźć w [4, 2].

LaTeX jest dostępny na większości platform sprzętowych, począwszy od PC i Macintosh, a skończywszy na dużych systemach wyposażonych w UNIX-a czy VMS. W wielu sieciach uniwersyteckich można spotkać gotowe do użytku instalacje LaTeX-a. Informacje, jak rozpocząć pracę w lokalnej instalacji LaTeX-a można znaleźć w *Local Guide* [5]. Jeżeli nie wiesz jak zacząć pracę z LaTeX-em to zapytaj się osoby, od której otrzymałeś to *Wprowadzenie*. W tym dokumencie nie poruszamy spraw związanych z instalowaniem i konfiguracją systemu LaTeX. Mówimy wyłącznie o tym, jak pisać dokumenty, aby mogły zostać przetwarzane przez LaTeX-a.

Wprowadzenie zawiera pięć następujących rozdziałów:

Rozdział 1 przedstawia ogólną strukturę dokumentów LaTeX-owych i omawia krótko historię systemu LaTeX. Po przeczytaniu tego rozdziału powinieneś mieć już wyobrażenie, czym jest LaTeX. Chociaż będzie to tylko obraz ogólny, pozwoli ci na połączenie informacji, jakie znajdziesz w pozostałych rozdziałach, w całościowy obraz.

Rozdział 2 podaje szczegóły dotyczące składanie dokumentów. Omawia najważniejsze instrukcje i środowiska. Po przeczytaniu tego rozdziału będziesz umiał tworzyć proste dokumenty LaTeX-owe.

Rozdział 3 poświęcono składaniu wzorów matematycznych. Wiele zamieszczonych tu przykładów pozwoli ci nauczyć się wykorzystywać jedną z najważniejszych możliwości LaTeX-a, jaką jest elegancki skład matematyki. Na końcu rozdziału zamieszczono zestawienie dostępnych w LaTeX-u symboli matematycznych.

Rozdział 4 wyjaśnia, jak tworzyć skorowidze i spisy bibliograficzne, jak dołączać rysunki w formacie EPS i jeszcze parę innych pożytecznych rzeczy.

Spis treści

Spis tabel

Spis literatury

[1] Carlisle David P., *Packages in the 'graphics' bundle.* Dokument dostępny w zestawie pakietów „graphics" w pliku `grfguide.tex`.

[2] Goossens Michel, Mittelbach Frank, Samarin Alexander. *The LaTeX Companion.* Addison-Wesley, Reading 1994, ISBN 0-201-54199-8.

[3] Knuth Donald E., *The TeXbook*, Addison-Wesley Publishing Company 1984, ISBN 0-201-13448-9.

[4] Lamport Leslie, *LaTeX: A Document Preparation System.* 2nd ed., Addison-Wesley, Reading 1994, ISBN 0-201-52983-1.

[5] *LaTeX Local Guide.* Każda wielodostępna instalacja LaTeX-owa powinna zawierać *LaTeX Local Guide*, w którym są opisane rzeczy specyficzne dla danej lokalnej instalacji. Dokument ten powinien być zawarty w pliku `local.tex`. W wielu wypadkach administratorzy nie udostępniają jednak użytkownikom takiego dokumentu. Pozostaje wtedy zwrócenie się o pomoc do lokalnego LaTeX-owego guru.

[6] LaTeX3 Project Team. *LaTeX 2_ε for authors.* Dokument dostępny w pliku `usrguide.tex` w dystrybucji LaTeX 2_ε.

[7] LaTeX3 Project Team. *LaTeX 2_ε for Class and Package writers.* Dokument dostępny w pliku `clsguide.tex` w dystrybucji LaTeX 2_ε.

[8] LaTeX3 Project Team. *LaTeX 2_ε Font selection.* Dokument dostępny w pliku `fntguide.tex` w dystrybucji LaTeX 2_ε.

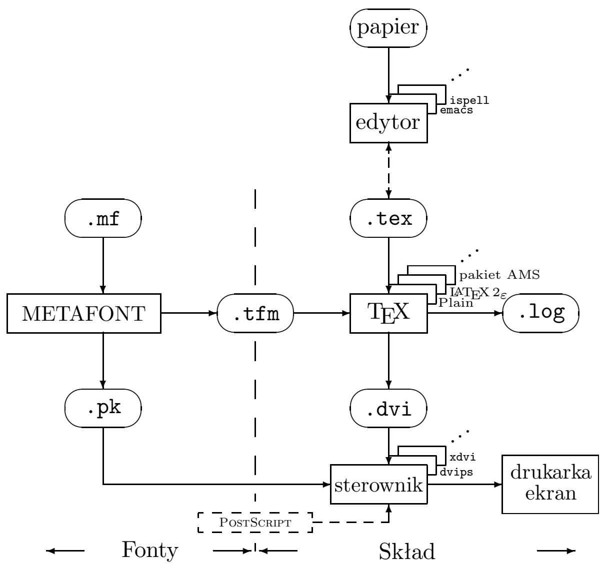
[9] Lichoński Bogusław, *TeX na indeksie.* Biuletyn GUST 1994 (3). Dokument dostępny w `http://www.gust.org.pl/PDF/BIUL/03/02-bl.pdf`.

[10] Nowacki Janusz M., *TeXnologia a typografia.* Biuletyn GUST 1995 (6). Dokument dostępny także w `http://www.gust.org.pl/PDF/BIUL/06/01-jmn.pdf`.

[11] Reckdahl Keith, *Using EPS Graphics in LaTeX 2_ε Documents* (szczegółowe wyjaśnienie jak dołączać grafikę w formacie EPS do dokumentów LaTeX-owych). Dostępny w `CTAN://tex-archive/info/epslatex.ps`.

[12] Schöpf Rainer, Raichle Bernd, Rowley Chris, *A New Implementation of LaTeX's verbatim Environments.* Dokument dostępny w zestawie pakietów „tools" w pliku `verbatim.dtx`.

[13] Williams Graham, *The TeX Catalogue* (katalog pakietów dla TeX-a oraz LaTeX-a). Dokument dostępny w `CTAN://tex-archive/help/Catalogue/catalogue.html`.



Prostokąty na rysunku oznaczają programy, a owale – pliki. W szczególności, z papierowego dokumentu za pomocą edytora (np. `emacs`) tworzymy dokument źródłowy (plik z rozszerzeniem `.tex`). Z dokumentu źrółowego po przetworzeniu LaTeX-em otrzymujemy plik `.dvi`, który jest z kolei plikiem wejściowym dla sterowników potrafiących go wydrukować bądź zamienić na inny format, np. PostScript. LaTeX zapisuje do pliku `.log` komunikaty podczas przetwarzania dokumentu. Pliki z rozszerzeniem `.tfm`, które *czyta* LaTeX zawierają informacje o fontach wykorzystywanych do składu.

Rysunek 1.1: Składniki systemu TEX

Kilka lat temu pakiet LaTeX został rozszerzony przez tak zwaną drużynę LaTeX3 (*LaTeX3 team*), kierowaną przez Franka Mittelbacha. Celem tego rozszerzenia było wprowadzenie kilku od dawna postulowanych ulepszeń oraz unifikacja rozmaitych odmian LaTeX-a, jakie rozpowszechniły się od chwili powstania kilkanaście lat temu LaTeX-a w wersji 2.09. Nowa wersja pakietu nazywa się LaTeX 2ε w celu odróżnienia jej od wersji poprzednich. Niniejszy dokument opisuje właśnie LaTeX 2ε.

Słowo LaTeX należy wymawiać „lej-tech” albo „la-tech”. Jeżeli nie można zapisać symbolu LaTeX, można zamiennie używać wersji `LaTeX`. LaTeX 2ε wymawiamy „lej-tech tu i” a zamienną wersją nazwy jest `LaTeX2e`.

Rysunek 1.1 pokazuje związek pomiędzy programem TEX i pakietem LaTeX 2ε. Rysunek zaczerpnięto z pliku `wots.tex` (autor Kees van der Laan).

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 znowu można skorzystać z poleceń `\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.
```

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

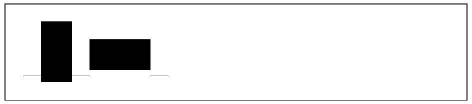
5.7. Kreski i podpory

Parę stron wcześniej pojawiło się polecenie:

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

W wyniku wykonania tej instrukcji najczęściej otrzymujemy w składzie czarny prostokąt:

```
\rule{3mm}{.1pt}%
\rule[-1mm]{5mm}{1cm}%
\rule{3mm}{.1pt}%
\rule[1mm]{1cm}{5mm}%
\rule{3mm}{.1pt}
```



Polecenia `\rule` używamy do rysowania kresek pionowych i poziomych. Na przykład gruba czarna krecha na stronie tytułowej niniejszego *Wprowadzenia* to wynik zadziałania instrukcji:

`\rule[-1ex]{\textwidth}{5pt}`

Parametr *przesunięcie* określa, jak wysoko przesunąć krechę ponad linię podstawową (lub opuścić poniżej linii podstawowej, jeżeli parametr jest ujemny).

Specjalnym przypadkiem jest kreska o zerowej szerokości, ale niezerowej wysokości. Taką kreskę nazywamy *podporą* (*strut*). Podpora to często stosowana metoda nadawania wszystkim elementom jednakowej wysokości.

- stopień pisma oraz numerację rozdziałów, podrozdziałów i punktów należy ustalić tak, aby czytelnik mógł się szybko zorientować w strukturze dokumentu;
- szerokość szpalty nie powinna być zbyt mała, aby nie zmuszać czytelnika do wytężania wzroku, ale zarazem wystarczająco duża, aby tekst elegancko wypełniał stronę.

W systemach wizualnych często powstają dokumenty przyjemne dla oka, ale pozbawione struktury, albo wykazujące brak konsekwencji w strukturze. LaTeX zapobiega powstawaniu takich błędów, zmuszając autora do określenia *logicznej* struktury dokumentu. Następnie LaTeX wybiera najodpowiedniejszy układ graficzny.

1.2.3. Zalety i wady

Tematem często dyskutowanym, gdy użytkownicy programów typu WYSIWYG spotykają użytkowników LaTeX-a, są „zalety LaTeX-a w porównaniu ze zwykłym procesorem tekstu” albo odwrotnie. Najlepiej podczas takich dyskusji siedzieć cicho. Czasami jednak nie uda się uniknąć zabrania głosu.

Poniżej zamieszczamy listę argumentów, które mogłyby być wykorzystane w takiej dyskusji. Oto największe zalety LaTeX-a w porównaniu ze zwykłymi procesorami tekstu.

- Dostępne są gotowe, przygotowane przez zawodowców układy graficzne, dzięki zastosowaniu których dokumenty wyglądają „jak z drukarni”.
- Wygodnie składa się wzory matematyczne.
- Do rozpoczęcia pracy wystarczy poznać zaledwie kilkanaście łatwych do zrozumienia instrukcji, określających strukturę logiczną dokumentu. Nie trzeba zaprzątać sobie głowy formatowaniem dokumentu.
- Nawet takie elementy jak przypisy, odnośniki, spisy treści, spisy tabel, skorowidze, oraz spisy bibliograficzne przygotowuje się bardzo łatwo.
- Istnieje wiele bezpłatnych pakietów poszerzających typograficzne możliwości LaTeX-a. Przykładowo, dostępne są 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 *The LaTeX Companion* [2].
- LaTeX zachęca autorów do tworzenia dokumentów o dobrze określonej strukturze.
- TeX – program formatujący używany przez LaTeX 2$_\varepsilon$ – jest bezpłatny i w najwyższym stopniu przenośny. Dzięki temu można działać na praktycznie każdej platformie systemowo-sprzętowej.

LaTeX ma także pewne braki, z tym że ciężko mi znaleźć chociaż jeden istotny. Jestem jednak pewien, że inne osoby wskażą ci ich setki;-)


```
\newenvironment{vardesc}[1]{%
\settowidth{\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.

5.6. Pudełka

LaTeX tworzy każdą stronę z pudełek, które odpowiednio skleja. Elementarnymi pudełkami są litery, z których sklejane są słowa. Słowa są następnie łączone w wiersze, a wiersze w akapity.

Trzeba przyznać, że powyższe jest mocno uproszczoną wersją tego, co się naprawdę dzieje, ale zasadniczo biorąc działanie LaTeX-a można wyjaśnić w terminach pudełek oraz kleju (odstępu wstawianego między pudełkami). Nie tylko litery są pudełkami. Do pudełka można włożyć praktycznie wszystko, także inne pudełka. Każde pudełko LaTeX traktuje jak pojedynczą literę.

Chciaż o tym nie mówiliśmy wprost, pudełka pojawiały się już w poprzednich rozdziałach. Na przykład polecenie `includegraphics` albo środowisko `tabular` tworzą pudełka. Oznacza to, że łatwo można zestawić dwa rysunki albo tabele obok siebie. Trzeba jedynie zadbać o to, by łączna szerokość połączonych obiektów nie była większa niż szerokość szpalty.

To samo odnosi się do akapitów, które – jeśli tego potrzebujemy – możemy składać w pudełka o zadanej szerokości:

`\parbox[`*pos*`]{`*szerokość*`}{`*tekst*`}`

Do tego samego celu można wykorzystać środowisko:

`\begin{minipage}[`*pos*`]{`*szerokość*`}` *tekst* `\end{minipage}`

Argument *pos* jest jednoliterowy i może przyjmować jedną z wartości: `c`, `t` lub `b`. Wartości te określają, jak LaTeX ma umieścić pudełko względem otaczającego tekstu. Wartość `c` oznacza umieszczenie środka wysokości pudełka na linii podstawowej, `t` – umieszczenie linii podstawowej pierwszego wiersza w pudełku na linii podstawowej otaczającego tekstu, natomiast `b` umieszczenie dolnej krawędzi pudełka na linii podstawowej. Wynik zastosowania parametrów ilustruje poniższy przykład (linię podstawową oznaczono kreską). Argument *szerokość* to wymiar określający szerokość pudełka.

Znaku w-tył-ciach *nie można* wstawić do tekstu metodą podwojenia. Kombinacja `\\` jest poleceniem LaTeX-a, opisanym w punkcie 2.2.1. Znak w-tył-ciach możemy wstawić za pomocą polecenia `$\backslash$`. Uwaga: znaki dolara są niezbędne, a ich opuszczenie spowoduje błąd podczas przetwarzania.

1.3.3. Polecenia LaTeX-a

Polecenia LaTeX-a występują w dwóch następujących odmianach:

1. Instrukcje rozpoczynające się znakiem w-tył-ciach „\", po którym występuje ciąg liter. Końcem instrukcji jest odstęp lub inny znak nie będący literą. W nazwach instrukcji LaTeX rozróżnia litery małe i duże. Używanie polskich liter diakrytycznych w nazwach instrukcji nie jest możliwe.
2. Instrukcje składające się ze znaku w-tył-ciach oraz *dokładnie jednego* znaku nie będącego literą.

Znaki odstępu po instrukcjach typu 1 są przez LaTeX-a ignorowane. Jeżeli po instrukcji ma w dokumencie występować odstęp, to należy bezpośrednio po niej umieścić kolejno: parę nawiasów klamrowych `{}` i odstęp. Para znaków `{}` zapobiega zignorowaniu przez LaTeX-a odstępu po nazwie instrukcji. Innym sposobem jest wstawienie specjalnej instrukcji `\ ` (tj. w-tył-ciach i spacja). Niektórzy używają jeszcze innego sposobu zapobiegania połykaniu spacji po nazwach instrukcji, a mianowicie wstawiają instrukcję pomiędzy parę nawiasów `{ i }`.

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

Czytałem, że Knuth dzieli użytkowników systemu TeX na TeXników oraz TeXpertów.
Dzisiaj jest 11 kwietnia 2000 roku.

Niektóre instrukcje LaTeX-owe mają argumenty. Argumenty podaje się w nawiasach klamrowych `{ }`, każdy argument w osobnej parze nawiasów. Liczba i kolejność argumentów jest dla każdego polecenia z góry ustalona. Instrukcje mogą mieć także argumenty opcjonalne (nieobowiązkowe), podawane w nawiasach kwadratowych `[ ]`. W przypadku większej liczby argumentów opcjonalnych oddziela się je przecinkami. Kolejność argumentów opcjonalnych *nie odgrywa roli*.

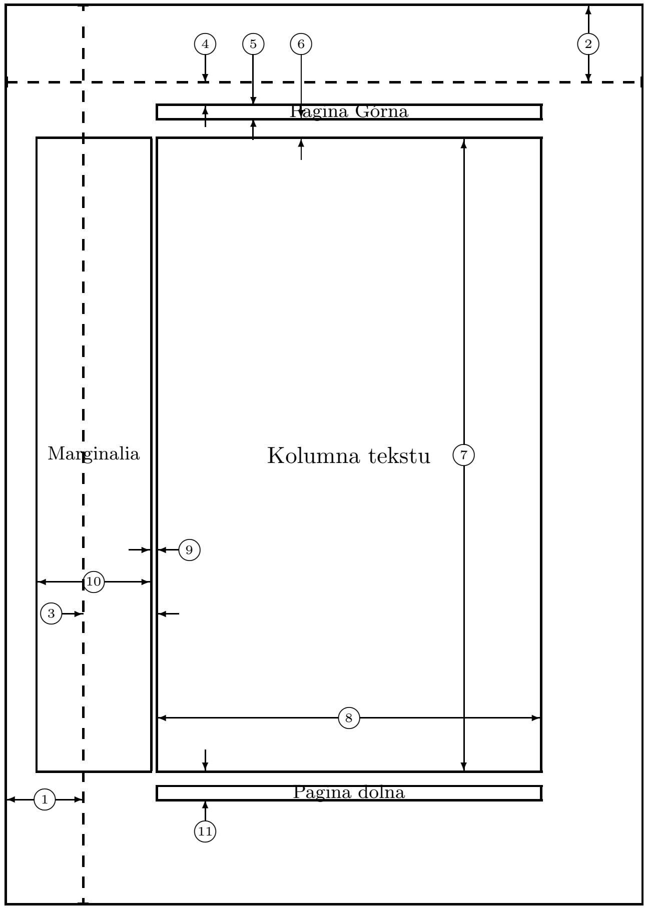
Poniższe przykłady ilustrują postać instrukcji LaTeX-owych. Znaczenie tych instrukcji jest na razie nieistotne; zostanie podane później.

```
Możesz na mnie \textsl{polegać}!

Tu wstaw zmianę wiersza.\newline
Dziękuję.
```

Możesz na mnie *polegać*!

Tu wstaw zmianę wiersza.
Dziękuję.



1	1 cal+ \hoffset	2	1 cal+ \voffset
3	\evensidemargin = 70pt	4	\topmargin = 22pt
5	\headheight = 12pt	6	\headsep = 19pt
7	\textheight = 595pt	8	\textwidth = 360pt
9	\marginparsep = 7pt	10	\marginparwidth = 106pt
11	\footskip = 27pt		\marginparpush = 5pt (nie pokazane)
	\hoffset = 0pt		\voffset = 0pt
	\paperwidth = 597pt		\paperheight = 845pt

Rysunek 5.2: Parametry układu graficznego strony

```
\documentclass[a4paper,11pt]{article}
\usepackage{latexsym}
\author{H.~Partl}
\title{Minimalism}
\frenchspacing
\begin{document}
\maketitle \tableofcontents
\section{Start}
Well and here begins my lovely article.
\section{End}
\ldots{} and here it ends.
\end{document}
```

Rysunek 1.3: Przykład artykułu

1.5. Układ graficzny dokumentu

1.5.1. Klasy dokumentów

LaTeX musi wiedzieć do jakiej klasy (typu) należy przetwarzany dokument. Jest to określone w instrukcji `\documentclass`:

`\documentclass[`*opcje*`]{`*klasa*`}`

gdzie *klasa* oznacza typ dokumentu. W tabeli 1.1 zestawiono klasy dokumentów opisane w niniejszym wprowadzeniu. W dystrybucji LaTeX 2_ε znajdują się także inne klasy. *Opcje* umożliwiają modyfikowanie klas. Poszczególne opcje rodziela się przecinkami. W tablicy 1.2 zestawiono najczęściej wykorzystywane opcje dla standardowych klas.

Tabela 1.1: Klasy dokumentów

`article`	artykuły, krótkie opracowania ...
`report`	dłuższe opracowania, dysertacje magisterskie i doktorskie ...
`book`	książki.
`slides`	przeźrocza[a].
`letter`	listy.

[a] Przeźrocza składane są bezszeryfowym krojem pisma. Zamiast klasy standardowej można do tworzenia slajdów wykorzystać pakiet FoilTeX `CTAN://tex-archive/macros/latex/packages/supported/foiltex`.

odstęp między wyrazami, przez odstęp pionowy rozumiemy odstęp między wierszami i akapitami) wstawiamy za pomocą polecenia:

`\hspace{`*odległość*`}`

Jeżeli taki odstęp, w wyniku złamania akapitu na wiersze, wypadnie na początku lub końcu wiersza, to zostanie usunięty – po to, by akapit nie był justowany „w chorągiewkę”. Jeżeli LaTeX ma wstawić odstęp także w wypadku, gdy rozpoczyna sie lub kończy wiersz, to zamiast `\hspace` należy użyć „gwiazdkowej” wersji `\hspace*`. Argument *odległość* oznacza wymiar LaTeX-owy. W najprostszej postaci jest to liczba oraz jednostka odległości. Wykaz ważniejszych dostępnych w LaTeX-u jednostek odległości znajduje się w tabeli 5.5.

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

To jest   odstęp równy 1,5 cm.

Tabela 5.5: LaTeX-owe 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 „elastycznej” odległości, zostawiając LaTeX-owi w pewnym zakresie swobodę wyboru 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*, mogą się kurczyć lub rozciągać w zakresie od $n - m$ do $n + p$. Omawiany wcześniej odstęp między akapitami (`\parskip`) jest przykładem wymiaru o zmiennej wielkości. Część wymiarów LaTeX-owych może mieć zmienne wielkości, ale część musi mieć wartość stałą. Na przykład, jest zrozumiałe, że wcięcie akapitowe musi być wielkością stałą, podobnie jak szerokość i wysokość łamu.

Polecenie:

`\stretch{`*n*`}`

wstawia specjalny odstęp rozciągliwy o tej właściwości, że potrafi wypełnić całą wolną przestrzeń wiersza. Jeżeli w linii wstawimy dwa polecenia

Tabela 1.3: Wybrane pakiety z podstawowej dystrybucji LaTeX-a

doc	Służy do drukowania dokumentacji pakietów oraz innych części składowych LaTeX-a. Opis znajduje się w pliku `doc.dtx`[a].
exscale	Umożliwia skalowanie fontów matematycznych, tak aby były optycznie zgodne z otaczającym tekstem, np. w tytułach rozdziałów. Opis w `ltexscale.dtx`.
fontenc	Definiuje układ znaków w bieżącym zestawie znaków (foncie). Opis w `ltoutenc.dtx`.
ifthen	Umożliwia korzystanie z poleceń typu `if... then do... otherwise do`. Opis w `ifthen.dtx` i [2].
latexsym	Udostępnia specjalny font symboliczny (fonty `lasy`). Opis w `latexsym.dtx` i [2].
makeidx	Udostępnia polecenia do przygotowywania skorowidzów. Opis w punkcie 4.3 i [2].
syntonly	Przetwarza dokument bez składania czegokolwiek. Przydatny do szybkiego sprawdzenia czy dokument nie zawiera błędów. Opis w `syntonly.dtx` i [2].
inputenc	Definiuje układ znaków w pliku źródłowym, jak: ASCII, ISO Latin-1, ISO Latin-2, 437/850 IBM, Apple Macintosh, Next, ANSI-Windows albo układ zdefiniowany przez użytkownika. Opis w `inputenc.dtx`.

[a] Plik ten powinien być zainstalowany w twoim systemie. Aby otrzymać plik `dvi` wystarczy napisać `latex doc.dtx` w katalogu z prawem do zapisu. To samo stosuje się do innych pakietów z tej tabeli.

gdzie *pakiet* oznacza nazwę pakietu, a *opcje* – listę oddzielonych przecinkami opcji. Część pakietów znajduje się w podstawowej dystrybucji LaTeX 2_ε (zobacz tabela 1.3). Pozostałe są rozpowszechniane oddzielnie. Gdy używamy LaTeX-a w systemie, którym zarządza (dobry) administrator, to informacja o dostępnych pakietach powinna się znajdować w *Local Guide* [5]. Podstawowym źródłem informacji o LaTeX-u jest *The LaTeX Companion* [2]. Zawiera on opis setek pakietów a także opisuje, jak rozszerzać możliwości LaTeX-a.

1.5.3. Style strony

Typowa strona składa się z trzech podstawowych części. Powyżej kolumny tekstu głównego znajduje się *pagina górna* (główka), która może zawierać numer strony, tytuł rozdziału czy punktu. Poniżej kolumny tekstu znajduje się *pagina dolna* (stopka). W niniejszym wprowadzeniu pagina dolna jest pu-

Tabela 5.4: Polecenia wyboru fontów w trybie matematycznym

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

stawową ideą LaTeX-a, jaką jest oddzielenie formy od treści dokumentu i posługiwanie się formatowaniem logicznym, a nie wizualnym. Jeżeli fragment tekstu ma zostać wyróżniony poprzez złożenie go innym krojem lub stopniem pisma, to należy zdefiniować odpowiednie polecenie i jego właśnie używać w treści dokumentu.

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

Baczność. 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 inny sposób 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: nie należy przesadzać z wykorzystywaniem wielu różnych krojów pisma w dokumencie.

5.3. Odstępy

5.3.1. Zmiana wielkości interlinii

Wielkość interlinii (odstępu pomiędzy wierszami) w dokumencie można zmieniać za pomocą polecenia:

`\linespread{`*czynnik*`}`

Spośrod instrukcji `\include` zostaną wykonane tylko te, dla których nazwy plików są na liście plików polecenia `\includeonly`. Uwaga: w liście plików nie można używać odstępów!

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

```
\input{plik}
```

która po prostu wstawia zawartość podanego pliku bez żadnych dodatkowych efektów.

Tabela 5.1: Polecenia wyboru krojów i odmian

`\textrm{...}`	krój szeryfowy	`\textsf{...}`	krój bezszeryfowy
`\texttt{...}`	`grotesk`		
`\textmd{...}`	pismo jasne	`\textbf{...}`	**pismo grube**
`\textup{...}`	odmiana prosta	`\textit{...}`	*kursywa*
`\textsl{...}`	*odmiana pochyła*	`\textsc{...}`	KAPITALIKI
`\emph{...}`	*wyróżnienie*		
`\textnormal{...}`	główny font dokumentu		

Tabela 5.2: Polecenia wyboru stopnia pisma i interlinii

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

Wewnątrz trybu matematycznego można wykorzystać instrukcje zmiany fontu do chwilowego opuszczenia trybu matematycznego i złożenia tekstu normalnym krojem.

Przy okazji omawiania poleceń dotyczących fontów trzeba wspomnieć o koncepcji *grupowania tekstu*. Grupa rozpoczyna się od znaku „{”, a kończy znakiem „}”. Grupy służą do ograniczania zasięgu działania poleceń LaTeX-a. 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. Polecenie `\small` zmienia stopień pisma na mały. W tym stopniu złożone jest tylko słowo „małe”, ponieważ nawias „}” 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 spowoduje, że resztę tekstu LaTeX złoży w wyjściowym stopniu pisma. Jak widać, grupy można zagnieżdżać (nawet wielokrotnie).

w tekście pojawiają się równania. 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 podrozdziału i przeczytać go jeszcze raz).

```
% Przykład 1
\ldots Słynne równanie Einsteina
\begin{equation}
e = m \cdot c^2 \; ,
\end{equation}
jest zarówno najbardziej znanym, ale także
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}

Natomiast, napięciowe prawo Kirchhoffa ma
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, który pojawia się po kropce kończącej zdanie, jest większy niż ten, który pojawia się 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 także części zdań. Wiele języków ma bardzo skomplikowaną interpunkcję, ale w większości przypadków (wliczając angielski i niemiecki), stawiając przecinek w określonym miejscu w zdaniu, na ogół nie popełnimy błędu, jeśli będziemy pamiętać o zasadzie, że przecinek oznacza przerwę/przystanek w zdaniu. Dlatego – jeśli nie jesteśmy pewni, gdzie w zdaniu należy go postawić – przeczytajmy zdanie na głos i postawmy przecinki wszędzie tam, gdzie zrobiliśmy krótką przerwę na wdech. Ale nie sugerujmy się tylko tym! Jeśli w danym miejscu przeci-


```
\newcommand{\wle}[1]
{\emph{#1} wprowadzenie
do systemu \LaTeXe}
% następnie po \begin{document}:
\wle{Krótkie}; \wle{Długie}
```

Krótkie wprowadzenie do systemu LaTeX 2_ε; *Długie* wprowadzenie do systemu LaTeX 2_ε

LaTeX nie pozwala definiować instrukcji, której nazwa, jest nazwą zdefiniowanego już polecenia. W wypadku, gdy chcemy zmienić znaczenie już istniejącej instrukcji, należy użyć polecenia `\renewcommand`. Za wyjątkiem nazwy ma ono składnię identyczną jak `\newcommand`. Czasami może się przydać polecenie `\providecommand`. Działa ono jak `\newcommand` z tym, że jeśli istnieje już polecenie o takiej samej nazwie, to nie zastępuje ono starej i zwyczajnie ignoruje nową.

5.1.2. Środowiska definiowane przez użytkownika

Odpowiednikiem instrukcji `\newcommand`, definiującej nowe polecenia, jest dla środowisk instrukcja `\newenvironment`. Ma ona następującą składnię:

`\newenvironment{`*nazwa*`}[`*num*`]{`*początek*`}{`*koniec*`}`

Podobnie jak w wypadku `\newcommand`, instrukcji `\newenvironment` można użyć z argumentem opcjonalnym lub bez. LaTeX wstawia tekst *początek* w miejsce `\begin{`*nazwa*`}`. Natomiast tekst *koniec* jest wstawiany w miejsce `\end{`*nazwa*`}`.

Poniższy przykład ilustruje sposób użycia instrukcji `\newenvironment`.

```
\newenvironment{zrodlo}
{Źródło: }{\par}
% następnie po \begin{document}:
\begin{zrodlo} Rocznik Statystyczny
GUS. \end{zrodlo}
```

Źródło: Rocznik Statystyczny GUS.

Znaczenie argumentu *num* jest takie samo jak w instrukcji `\newcommand`.

LaTeX nie pozwala zdefiniować środowiska o już istniejącej nazwie. W wypadku, gdy chcemy zmienić znaczenie już istniejącego środowiska, należy użyć polecenia `\renewenvironment`, mającego taką samą składnię jak `\newenvironment`.

5.1.3. Własne pakiety

W wypadku zdefiniowania wielu nowych środowisk i poleceń preambuła dokumentu może się znacznie wydłużyć. Dobrze w takiej sytuacji jest stworzyć pakiet zawierający własne instrukcje i definicje środowisk. Taki pakiet można dołączyć do dokumentu za pomocą polecenia `\usepackage`.

Instrukcje:

`\linebreak[n]`, `\nolinebreak[n]`, `\pagebreak[n]` oraz `\nopagebreak[n]`

oznaczają – odpowiednio – zachętę do złamania wiersza, niezgodę na złamanie wiersza, zachętę do złamania strony, niezgodę na złamanie strony, w miejscu ich wystąpienia w dokumencie. Opcjonalny argument n, o dopuszczalnej wartości od 0 do 4, określa stopień tej zachęty (niezgody). Domyślna wartość 4, to bezwarunkowy zakaz lub nakaz złamania linii/strony. Wartości mniejsze od 4 pozostawiają LaTeX-owi swobodę zignorowania instrukcji, jeżeli skład otrzymany w rezultacie byłby niskiej jakości.

Poleceń z grupy `break` nie należy mylić z poleceniami z grupy `new`. Mimo wydania polecenia typu `break` LaTeX stara się wypełnić wiersz do prawego skraju czy też wypełnić stronę do całej jej wysokości. Nietrudno zgadnąć, jakiej instrukcji należy użyć, gdy naprawdę zależy nam na rozpoczęciu nowego wiersza (ta zagadka jest łatwa pod warunkiem, że znamy język angielski – jest to polecenie `\newline`).

Jeżeli w wierszu zakończonym instrukcją `\newline` jest zbyt mało tekstu, to LaTeX nie wyrówna tego wiersza do prawego marginesu, tylko po tekście wstawi odpowiedni odstęp. Jeżeli zamiast `\newline` użyjemy `\linebreak`, to LaTeX postara się wyrównać zawartość kończonej linijki do prawego marginesu. Zbyt małe wypełnienie wiersza tekstem zmusi LaTeX-a do wstawienia zbyt dużych odstępów pomiędzy wyrazy w wierszu, co zostanie zasygnalizowane podczas przetwarzania wyświetleniem komunikatu:

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

W komunikacie tym liczba po słowie `badness` wskazuje, jak bardzo LaTeX-owi „nie podoba" się złożony wiersz. W tym przypadku jest to maksymalna wartość, 10000. Liczby na końcu komunikatu to numery pierwszej i ostatniej linii akapitu, w którym LaTeX musiał złożyć wiersz z nadmiernymi odstępami między wyrazami. Chociaż wielu użytkowników LaTeX-a nie zwraca uwagi na podobne ostrzeżenia, warto zdawać sobie sprawę, co one oznaczają.

Oprócz wyżej wymienionych istnieją w LaTeX-u jeszcze instrukcje:

`\clearpage`, `\cleardoublepage`

Obydwie rozpoczynają skład nowej strony. Instrukcja `\cleardoublepage` działa tak, jak `\clearpage`, z tym że rozpoczynana nowa strona jest zawsze nieparzysta. W razie potrzeby tworzona jest strona pusta. W składzie dwuszpaltowym (opcja `twocolumn`) instrukcja `\newpage` kończy szpaltę natomiast `\clearpage` oraz `\cleardoublepage` kończą stronę, w miarę potrzeby pozostawiając pustą prawą szpaltę.

Jeżeli na stronie zakończonej instrukcją `\newline` (albo `\clearpage`) jest zbyt mało tekstu, LaTeX wstawia odpowiedni odstęp u dołu strony, a nie pomiędzy akapity czy też inne elementy na stronie, tak jak ma to miejsce w przypadku instrukcji `\pagebreak`. Jeżeli te odstępy okazują się za duże (a zdarza się to dość często), to sytuacja taka jest sygnalizowana podczas przetwarzania dokumentu odpowiednim ostrzeżeniem, np.:

```
Underfull \vbox (badness 10000) has occurred
while \output is active [7]
```

Rozdział 5

Adaptowanie LaTeX-a

Dokumenty składane z wykorzystaniem poznanych do tej pory poleceń zapewne będą się podobać zdecydowanej większości czytelników. Chociaż ich wygląd nie będzie może wyrafinowany, z pewnością jednak spełnią podstawowe zasady składu, dzięki czemu łatwo i przyjemnie będzie się je czytało.

W niektórych wypadkach może się jednak okazać, że brakuje polecenia czy środowiska, za pomocą którego moglibyśmy złożyć dany fragment tekstu w sposób odpowiadający potrzebom, albo też, że sposób działania dostępnej w LaTeX-u instrukcji nie spełnia naszych wymagań.

W tym rozdziale przedstawimy, jak nauczyć LaTeX-a formatować dokumenty tak, aby wyglądały inaczej, niż w przypadku korzystania tylko ze standardowych klas i pakietów.

5.1. Definiowane instrukcji i środowisk

Czytelnicy zapewne zauważyli, że nowo wprowadzane w tej książce polecenia ukazują się w ramkach oraz, że znajdują się w skorowidzu. Aby to osiągnąć, nie korzystaliśmy za każdym razem z wbudowanych w LaTeX-a instrukcji, tylko utworzyliśmy własny pakiet, w którym zawarliśmy nowe, potrzebne nam polecenia i środowiska. Mając taki pakiet, wystarczy po prostu napisać:

```
\begin{command}
\ci{polecenie}
\end{command}
```

`\polecenie`

W tym przykładzie użyliśmy zarówno nowego środowiska o nazwie `command`, odpowiedzialnego za rysowanie ramek dookoła instrukcji, jak też nowego polecenia `\ci`, służącego do składu nazw poleceń i wprowadzania ich do skorowidza. Proponujemy Czytelnikom odszukanie hasła `\polecenie` w skorowidzu; przy haśle powinny być podane numery stron, na których ta instrukcja występuje w książce.

w ogóle. Wyrazy z listy nie mogą zawierać żadnych znaków specjalnych ani symboli.

Przykład:

```
\hyphenation{ćwierć-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. Co więcej, próba wykorzystania polecenia `\hyphenation` w przypadku plików źródłowych zapisanych w notacji prefixowej albo też plików w notacji naturalnej i korzystania z pakietu inputenc zakończy się błędem w czasie przetwarzania dokumentu. Ten fenomen jest spowodowany sposobem w jakim działa TEX, a jego wyjaśnienie daleko wykracza poza ramy niniejszego wprowadzenia.

Instrukcja „`\-`” wskazuje, w których miejscach LATEX-owi *wolno* przenieść wyraz w danym miejscu w dokumencie. LATEX nie podzieli słowa w innym miejscu. Instrukcja ta przydaje się szczególnie w przypadku wyrazów zawierających znaki specjalne (np. znaki akcentowane), ponieważ LATEX dzieli automatycznie tylko wyrazy złożone z samych liter.

```
Nie\-bie\-sko\-bia\-ło\-zie\-lo\-%
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}.

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

Numer mojego telefonu wkrótce się zmieni na 0116 291 2319.

Parametr *nazwa* to nazwa pliku.

2.3. Znaki specjalne i symbole

2.3.1. Cudzysłowy

Znaku cudzysłowu `"` używa się inaczej niż na maszynie do pisania, bo w publikacjach drukowanych inaczej oznacza się początek i inaczej koniec cudzysłowu. Występują także różnice w sposobach oznaczania cudzysłowów w różnych językach. Dwa apostrofy ` otwierają, a dwa apostrofy ' zamykają LATEX-owy cudzysłów według reguł języka angielskiego:

```
``Please press the `x' key.''
```

“Please press the ‘x’ key.”


```
\documentclass{book}
\usepackage{fancyhdr}
\pagestyle{fancy}
% zmiana liter w~żywej paginie 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

jest nadawane nowe znaczenie (nowe wartości) za każdym wykonaniem instrukcji składania tytułu rozdziału i punktu (`\chapter`, `\section`).

W rzeczywistości, `\chapter` oraz inne polecenia podziału logicznego dokumentu nie zmieniają definicji poleceń `\rightmark` i `\leftmark`. Odwołują się one natomiast do poleceń `\chaptermark`, `\sectionmark`, lub `\subsectionmark`. Dopiero użycie tych instrukcji powoduje zmianę definicji poleceń `\rightmark` i `\leftmark`.

Do zmiany postaci tytułu rozdziału w główce wystarczy modyfikacja polecenia `\chaptermark`. Rysunek 4.1 przedstawia takie wykorzystanie pakietu, że paginy górne będą wyglądać mniej więcej tak, jak w tym podręczniku. Kompletny opis pakietu znajduje się w jego dokumentacji dostępnej pod adresem podanym w ostatnim przypisie.

4.5. Pakiet `verbatim`

Pakiet `verbatim` udostępnia poprawioną wersję standardowego środowiska `verbatim` (opisanego na stronie 33). Oprócz wielu drobnych ale istotnych ulepszeń pakiet udostępnia polecenie:

`\verbatiminput{`*plik*`}`

2.3.3. Wielokropek (. . .)

W typowym piśmie maszynowym przecinek albo 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 miedzy nimi będą zbyt małe. Wielokropek (trzy kropki) uzyskujemy za pomocą instrukcji `\ldots`. Przykład:

```
\ldots
```

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

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

2.3.4. 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

LaTeX automatycznie wstawia ligatury zamiast odpowiednich sekwencji znaków. Można temu zapobiec wstawiając instrukcję `\mbox{}` pomiędzy znaki tworzące ligaturę.

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

Jak lepiej: geograﬁi czy geografii?

2.3.5. Akcenty i znaki specjalne

LaTeX umożliwia wstawienie znaków akcentowanych i specjalnych używanych w wielu językach. W tabeli 2.1 zestawiono wszystkie możliwe akcenty dla litery „o”. Instrukcje te można stosować również do dowolnej innej litery.

W przypadku umieszczenia akcentu nad literami „i” oraz „j”, należy usunąć kropkę znad tych liter. Do tego celu służą instrukcje `\i` i `\j`, wstawiające do składu specjalne wersje liter „ı” oraz „ȷ”.

```
H\^otel, na\"\i ve, \'el\`eve,\\
sm\o rrebr\o d, !`Se\~norita!,\\
Sch\"onbrunner Schlo\ss{}
Stra\ss e
```

Hôtel, naïve, élève,
smørrebrød, ¡Señorita!,
Schönbrunner Schloß Straße

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 do pliku `.aux` przez LaTeX-a, BibTeX tworzy spis literatury obejmujący tylko te pozycje z biblioteki (zwykle plik ten ma rozszerzenie `.bib`), które cytowano w dokumencie. Format spisu zależy od specyfikacji szablonu znajdującej się w pliku o rozszerzeniu `.bst` i zapisywany jest do pliku o rozszerzeniu `.bbl`. Do poprawnego sformatowania bibliografii i cytowań konieczne jest przynajmniej trzykrotne przetworzenie dokumentu LaTeX-em.

4.3. Skorowidze

Niezwykle użytecznym elementem wielu książek jest skorowidz. Można go bardzo łatwo utworzyć za pomocą LaTeX-a oraz programu wspomagającego o nazwie `makeindex`[4]. W tym wprowadzeniu omówimy jedynie podstawowe polecenia dotyczące przygotowania skorowidzy. Znowu więcej informacji znajdziemy w *The LaTeX Companion* [2].

Generowanie haseł do skorowidza jest możliwe dopiero po dołączeniu w preambule dokumentu pakietu o nazwie makeidx:

```
\usepackage{makeidx}
```

oraz wstawieniu (także w obrębie preambuły) instrukcji:

```
\makeindex
```

Każde hasło wstawiamy do skorowidza poleceniem:

```
\index{hasło}
```

gdzie *hasło* oznacza pozycję w skorowidzu. Polecenie `\index{`*hasło*`}` umieszczamy w pliku źródłowym bezpośrednio w miejscu związanym z określonym hasłem. W tabeli 4.2 przedstawiono przykłady użycia *haseł*.

Podczas przetwarzania pliku źródłowego przez LaTeX-a każda instrukcja `\index` powoduje zapisanie odpowiedniej pozycji skorowidza oraz numeru strony, którego ta pozycja dotyczy do pliku pomocniczego. Plik ten ma tę samą nazwę co główny plik źródłowy i rozszerzenie `.idx`. Z kolei plik `.idx` musi być przetworzony za pomocą programu `makeindex`, piszemy:

```
makeindex plik
```

[4]Lub `makeidx`, jeśli nasz system operacyjny nie pozwala używać nazw dłuższych niż 8 znaków.

Po odpowiednim skonfigurowaniu naszego systemu można wykorzystać pakiet babel. Wystarczy w tym celu użyć w preambule instrukcję:

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

Argument *lista-języków*, to oddzielone przecinkami nazwy języków, które obsługuje lokalna wersja LaTeX-a. Ostatni na liście jest językiem domyślnym. Podręcznik *Local Guide* [5] zawiera informacje, o tym jakie nazwy mogą się pojawić na *liście-języków*.

Dla niektórych języków babel udostępnia specjalne instrukcje, ułatwiające wprowadzanie znaków diakrytycznych i specjalnych. Przykładowo teksty w języku niemieckim zawierają dużo 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 od 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. dystrybucja LaTeX-a zawiera pakiet inputenc, umożliwiający kodowanie znaków diakrytycznych zgodnie z wieloma wariantami, np. ISO-8859-2 (Linux) lub CP 1250 (MS Windows). Więcej informacji na ten temat można znaleźć w dokumentacji pakietu. Wykorzystując pakiet musimy sobie zdawać sprawę, że część użytkowników nie będzie w stanie prawidłowo wyświetlić tak przygotowanych plików źródłowych. Przykładowo znak ą ma numer 161 w standardzie ISO-8859-2, podczas gdy w kodowaniu zgodnym z CP 1250 ta sama litera ma kod 169 (a znak o kodzie 161 oznacza jakiś inny „krzaczek").

Drugą stroną medalu jest układ znaków (kodowanie) fontu. Kodowanie fontu określa, w jakich miejscach *fontu* znajdują się poszczególne znaki. Oryginalne fonty Computer Modern (CM) są 128-znakowe i nie zawierają na przykład kompletu polskich znaków. Znaki diakrytyczne można tworzyć za pomocą połączenia dwóch znaków: litery i odpowiedniego akcentu. Taki sposób powoduje, że TeX nie może prawidłowo przenosić wyrazów zawierających znaki diakrytyczne, a także, że typograficzna jakość w ten sposób skonstruowanych diakrytyków jest kiepska.

Na szczęście prawie wszystkie współczesne dystrybucje TeX-a zawierają komplet fontów European Computer Modern (EC). Są to fonty ośmiobitowe zawierające 256 znaków. Pierwszych 128 znaków fontu EC są (prawie) identyczne ze znakami z fontów CM. Pozostałe 128 znaków to znaki diakrytyczne wykorzystywane w wielu językach europejskich, w tym komplet znaków niezbędnych do składania polskich tekstów. Fonty EC umożliwiają wykorzystanie pełnych możliwości TeX-owego mechanizmu przenoszenia wyrazów; znacznie lepsza jest też jakość typograficzna znaków diakrytycznych.

Ponadto istnieją jeszcze fonty PL (autorzy B. Jackowski i M. Ryćko). Są to fonty zgodne z CM zawierające dodatkowo wszystkie polskie znaki diakrytyczne. Były to

Nie mamy takiej możliwości wykorzystując interpreter w drukarce. Program ghostscript jest dostępny na wszystkich popularnych platformach systemowych.

Do dołączenia rysunków do dokumentu można skorzystać z poleceń pakietu graphicx (autor: D. P. Carlisle). Pakiet ten jest częścią zestawu pakietów zwanego „graphics"[2].

Zakładając, że dysponujemy interpreterem języka PostScript oraz pakietem graphicx, dołączanie grafiki można przedstawić następująco:

1. Zapisujemy rysunek w formacie EPS[3].
2. Dołączamy pakiet graphicx do preambuły dokumentu:

   `\usepackage[`*dvi-ps*`]{graphicx}`

   gdzie *dvi-ps* oznacza nazwę sterownika wykorzystywanego do zamiany pliku `.dvi` na plik postscriptowy. Najczęściej używanym do tego celu programem jest `dvips`. Nazwa sterownika jest tu konieczna z uwagi na brak standardu dotyczącego dołączania grafiki PostScriptowej do dokumentów LaTeX-owych. Podanie nazwy pozwala na wybór metody specyficznej dla danego sterownika.
3. Instrukcją:

   `\includegraphics[`*klucz*=*wartość*, ... `]{`*plik*`}`

   dołączamy *plik* do dokumentu. Parametr opcjonalny jest listą, oddzielonych przecinkami *kluczy* o określonych przez nas *wartościach*. Klucze wykorzystujemy do zmian parametrów rysunku, tj. szerokości czy wysokości rysunku oraz kąta obrotu. W tablicy 4.1 zamieszczono najważniejsze klucze.

Tabela 4.1: Nazwy kluczy w pakiecie graphicx

`width`	skalowanie rysunku do szerokości *width*;
`height`	skalowanie rysunku do wysokości *height*;
`angle`	obrót (przeciwnie do ruchu wskazówek zegara);
`scale`	skalowanie równomierne.

Poniższy przykład pozwoli nam lepiej zrozumieć to, co zostało przedstawione w powyższych punktach:

[2] `CTAN://tex-archive/macros/latex/packages/graphics`

[3] Jeżeli program graficzny, którym się posługujemy nie pozwala nam na to, spróbujmy zainstalować sterownik do drukarki postscriptowej w rodzaju Apple Laser Writer i drukować za pomocą tego sterownika do pliku. Przy odrobinie szczęścia plik wynikowy będzie w formacie EPS. Pamiętajmy jednak, że pliki w tym formacie mają tylko jedną stronę. Sterowniki niektórych drukarek można jawnie ustawić właśnie do tworzenia plików w tym formacie.

Ponadto zapis np. `\'o` nie oznacza – jak w LaTeX-u 2.09 – „wstaw akcent *acute* nad literę o”, tylko „użyj najlepszej metody uzyskania znaku *oacute* dostępnej w bieżącym układzie fontu”. Jeżeli w bieżącym foncie jest dostępny znak *oacute*, zostanie on użyty bezpośrednio. Jeżeli go nie ma, to TeX złoży taki znak z litery o oraz odpowiedniego akcentu[2].

Posługiwanie się powyższymi poleceniami do pisania tekstów po polsku jest uciążliwe. Stąd pomysł notacji prefiksowej; polega ona na zapisywaniu polskich znaków diakrytycznych w postaci dwóch znaków: ustalonego znaku specjalnego (prefiksu) i litery łacińskiej podobnej graficznie do odpowiedniej polskiej. Notacja prefiksowa ma wiele wersji: stosowano różne znaki jako prefiks (co najmniej `"`, `@` i `/`) oraz różne konwencje zapisywania ź oraz ż (przynajmniej jeden z nich musi być zapisywany przy pomocy znaku różnego od `z`).

Pakiet babel umożliwia zapis prefiksowy z wykorzystaniem znaku cudzysłowu, oto przykład zapisu minuskuł polskich znaków diakrytycznych w tej notacji (`"r` oznacza ż):

```
"a "c "e "l "n "o "s "z "r
```

Ponadto „`"‘`” oznacza polski cudzysłów otwierający, „`"<`” francuski cudzysłów otwierający a „`">`” francuski cudzysłów zamykający.

Bezpośrednie wprowadzanie polskich znaków umożliwia pakiet inputenc opisany w poprzednim punkcie. W zależności od platformy, na której pracujemy, mamy do wyboru ISO-8859-2 (Unix, Linux), CP1250 (Windows) lub inne, rzadziej spotykane strony kodowe. Przykładowo, jeżeli dokument jest kodowany w standardzie ISO-8859-2 (Latin-2) pakiet inputenc należy dołączyć do dokumentu następująco:

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

W przypadku, gdy dokument jest kodowany w standardzie CP 1250, zamiast opcji `latin2` powinniśmy wpisać `cp1250`.

Dodatkowo, w preambule dokumentu należy umieścić polecenie dołączenia pakietu fontenc z opcją `T1`, co spowoduje, że LaTeX będzie korzystał z 256 znakowych fontów EC. Fonty te zawierają dobrej jakości polskie znaki diakrytyczne.

Pozostał jeszcze do rozwiązania problem zapoznania LaTeX-a z polskimi regułami dzielenia wyrazów. Jeżeli posługujemy się LaTeX-em w środowisku wielodostępnym, na przykład na uczelni, to przypuszczalnie jest to już zrobione przez administratora, ale jeżeli rozpoczynamy pracę na własnym komputerze, to musimy niezbędną adaptację wykonać sami. W tym celu należy odszukać plik o nazwie `language.dat`. W pliku tym znajdują się informacje o dostępnych w pakiecie wzorcach dzielenia wyrazów. Interesujący nas fragment pliku wygląda mniej więcej tak:

```
%norsk nohyph.tex
%polish plhyph.tex
%portuges pthyph.tex
```

Pierwszy wyraz to nazwa języka (którą potem posługuje się LaTeX, np. jako argumentem polecenia `\selectlanguage`), drugi to *nazwa* pliku, w którym znajdują się reguły podziału wyrazów dla tego języka.

Jeśli chcemy dołączyć do LaTeX-a polskie wzorce przenoszenia, to powinniśmy usunąć znak procentu z linii zawierającej słowo `polish`. Na ogół LaTeX zna wzorce

[2] Z notacją tą wiąże się jednak dobrze znany problem: środowisko `tabbing` zmienia lokalnie definicję kilku makr, w tym `\'`. Dlatego w jego obrębie do uzyskiwania znaków z akcentem acute trzeba używać notacji typu `\a'o`. W konsekwencji zarówno w implementacji notacji prefiksowej, jak i „stron kodowych” pakietu inputenc do akcentu *acute* trzeba dostawać się nieco naokoło.

Tabela 3.18: Różne symbole (pakiet AMS)

$\hbar$	`\hbar`	$\hslash$	`\hslash`	$\Bbbk$	`\Bbbk`
$\square$	`\square`	$\blacksquare$	`\blacksquare`	$\circledS$	`\circledS`
$\vartriangle$	`\vartriangle`	$\blacktriangle$	`\blacktriangle`	$\complement$	`\complement`
$\triangledown$	`\triangledown`	$\blacktriangledown$	`\blacktriangledown`	$\Game$	`\Game`
$\lozenge$	`\lozenge`	$\blacklozenge$	`\blacklozenge`	$\bigstar$	`\bigstar`
$\angle$	`\angle`	$\measuredangle$	`\measuredangle`	$\sphericalangle$	`\sphericalangle`
$\diagup$	`\diagup`	$\diagdown$	`\diagdown`	$\backprime$	`\backprime`
$\nexists$	`\nexists`	$\Finv$	`\Finv`	$\varnothing$	`\varnothing`
$\eth$	`\eth`	$\mho$	`\mho`		

Tabela 3.19: Kroje pisma dostępne w trybie matematycznym

Przykład	Polecenie	Wymagany pakiet
ABCdef	`\mathrm{ABCdef}`	
ABCdef	`\mathit{ABCdef}`	
$\mathnormal{ABCdef}$	`\mathnormal{ABCdef}`	
$\mathcal{ABC}$	`\mathcal{ABC}`	
$\mathscr{ABC}$	`\mathcal{ABC}`	mathrsfs
$\mathcal{ABC}$	`\mathcal{ABC}`	eucal z opcją: mathcal albo
	`\mathscr{ABC}`	eucal z opcją: mathscr
$\mathfrak{ABCdef}$	`\mathfrak{ABCdef}`	eufrak
$\mathbb{ABC}$	`\mathbb{ABC}`	amsfonts albo amssymb

`OT1` – wybranie kodowania `OT1` fontów, co oznacza skład fontami CM (nie polecane);
`plmath` – zmienia definicje znaków mniejszy-równy i większy-równy[4];
`nomathsymbols` – blokuje zmianę znaczenia standardowych poleceń trygonometrycznych oraz relacji mniejszy-równy i większy-równy;
`MeX` – tryb pełnej polonizacji, w praktyce skrót dla opcji `OT4` + `plmath`.

Jeżeli nie wybrano żadnej z opcji układu kodowania, to platex usiłuje włączyć fonty PL, jeżeli są one zainstalowane. Dotyczy to zarówno fontów tekstowych jak i tych zawierających znaki matematyczne. W instalacji zawierającej fonty PL dołączenie pakietu bez opcji jest równoważne:

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

Polecenie `\selecthyphenation` pozwala przełączyć się na dany zestaw wzorców dzielenia wyrazów. Jest to odpowiednik polecenia `\selectlanguage` w pakiecie babel. Argumentem jest nazwa języka, taka jak w pliku `language.dat`.

Notację ciachową, która domyślnie jest *wyłączona*, można uaktywnić poleceniem `\prefixing`. Polecenie to powinno występować w preambule dokumentu dopiero po poleceniach dołączających pakiety.

Wprowadzanie polskich znaków w sposób bezpośredni wymaga dołączenia opisanego powyżej pakietu inputenc.

Instalowanie pakietu platex trzeba z reguły rozpocząć od jego skopiowania w odpowiednie miejsce do naszej instalacji LaTeX-owej ponieważ w przeciwieństwie do pakietu babel nie jest on dołączany do standardowej dystrybucji LaTeX-a (być może to się zmieni w przyszłości). Pakiet znajduje się w `CTAN://tex-archive/language/polish/platex-1.01.zip` lub `ftp://ftp.gust.org.pl/TeX/language/polish/platex-1.01.zip`. Miejsce, w które należy skopiować pakiet, zależy od instalacji. Dla systemów teTeX, web2c i MikTeX będzie to `.../texmf/tex/latex/`. Trzy kropki z przodu wskazują, że katalog `texmf/` nie musi się znajdować w katalogu głównym, tylko raczej w jakimś miejscu położonym „niżej" w strukturze katalogów. Po znalezieniu właściwego katalogu `texmf/tex/latex/` tworzymy tam podkatalog, `platex`, i rozpakowujemy w nim zawartość pakietu. Następnie wydajemy polecenie:

```
latex platex.ins
```

Do uzyskania poprawnego przenoszenia polskich słów konieczne jest wygenerowanie nowego formatu LaTeX-a. W tym celu należy usunąć (lub zmienić nazwy, tak aby zostały pominięte przez LaTeX-a) oryginalne pliki `hyphen.cfg`, `language.dat`. Ponadto w wielu starszych instalacjach plik `hyphen.tex` jest podmieniony „pod" potrzeby babel-a i nie zawiera angielskich wzorców podziału. W tym przypadku trzeba zastąpić plik `hyphen.tex` „prawdziwym" plikiem z angielskimi regułami dzielenia. Następnie z linii poleceń piszemy:

```
initex latex.ltx
```

Z tak powstałym plikiem `latex.fmt` postępujemy tak samo, jak przy tworzeniu nowego LaTeX-a dla pakietu babel.

[4] W systemie muszą być zainstalowane fonty PL. Są one dostępne w `CTAN://tex-archive/language/polish/pl-mf.zip`. Opcjonalnie można także skopiować pliki metryczne tych fontów. Są one dostępne w `CTAN://tex-archive/language/polish/pl-tfm.zip`. Pliki metryczne fontów, w przypadku wykorzystywania nowych instalacji TeX-a typu teTeX, web2c czy MikTeX nie są konieczne, gdyż jeżeli ich nie ma, są tworzone automatycznie. Nie będziemy omawiać szczegółowo instalacji fontów PL, gdyż sposób jej przeprowadzenia zależy mocno od konkretnej dystrybucji.

Tabela 3.14: Symbole relacji (pakiet AMS)

$\lessdot$	`\lessdot`	$\gtrdot$	`\gtrdot`	$\doteqdot$	`\doteqdot` albo `\Doteq`
$\leqslant$	`\leqslant`	$\geqslant$	`\geqslant`	$\risingdotseq$	`\risingdotseq`
$\eqslantless$	`\eqslantless`	$\eqslantgtr$	`\eqslantgtr`	$\fallingdotseq$	`\fallingdotseq`
$\leqq$	`\leqq`	$\geqq$	`\geqq`	$\eqcirc$	`\eqcirc`
$\lll$	`\lll` albo `\llless`	$\ggg$	`\ggg` albo `\gggtr`	$\circeq$	`\circeq`
$\lesssim$	`\lesssim`	$\gtrsim$	`\gtrsim`	$\triangleq$	`\triangleq`
$\lessapprox$	`\lessapprox`	$\gtrapprox$	`\gtrapprox`	$\bumpeq$	`\bumpeq`
$\lessgtr$	`\lessgtr`	$\gtrless$	`\gtrless`	$\Bumpeq$	`\Bumpeq`
$\lesseqgtr$	`\lesseqgtr`	$\gtreqless$	`\gtreqless`	$\thicksim$	`\thicksim`
$\lesseqqgtr$	`\lesseqqgtr`	$\gtreqqless$	`\gtreqqless`	$\thickapprox$	`\thickapprox`
$\preccurlyeq$	`\preccurlyeq`	$\succcurlyeq$	`\succcurlyeq`	$\approxeq$	`\approxeq`
$\curlyeqprec$	`\curlyeqprec`	$\curlyeqsucc$	`\curlyeqsucc`	$\backsim$	`\backsim`
$\precsim$	`\precsim`	$\succsim$	`\succsim`	$\backsimeq$	`\backsimeq`
$\precapprox$	`\precapprox`	$\succapprox$	`\succapprox`	$\vDash$	`\vDash`
$\subseteqq$	`\subseteqq`	$\supseteqq$	`\supseteqq`	$\Vdash$	`\Vdash`
$\Subset$	`\Subset`	$\Supset$	`\Supset`	$\Vvdash$	`\Vvdash`
$\sqsubset$	`\sqsubset`	$\sqsupset$	`\sqsupset`	$\backepsilon$	`\backepsilon`
$\therefore$	`\therefore`	$\because$	`\because`	$\varpropto$	`\varpropto`
$\shortmid$	`\shortmid`	$\shortparallel$	`\shortparallel`	$\between$	`\between`
$\smallsmile$	`\smallsmile`	$\smallfrown$	`\smallfrown`	$\pitchfork$	`\pitchfork`
$\vartriangleleft$	`\vartriangleleft`	$\vartriangleright$	`\vartriangleright`	$\blacktriangleleft$	`\blacktriangleleft`
$\trianglelefteq$	`\trianglelefteq`	$\trianglerighteq$	`\trianglerighteq`	$\blacktriangleright$	`\blacktriangleright`

Tabela 3.15: Strzałki (pakiet AMS)

$\dashleftarrow$	`\dashleftarrow`	$\dashrightarrow$	`\dashrightarrow`	$\multimap$	`\multimap`
$\leftleftarrows$	`\leftleftarrows`	$\rightrightarrows$	`\rightrightarrows`	$\upuparrows$	`\upuparrows`
$\leftrightarrows$	`\leftrightarrows`	$\rightleftarrows$	`\rightleftarrows`	$\downdownarrows$	`\downdownarrows`
$\Lleftarrow$	`\Lleftarrow`	$\Rrightarrow$	`\Rrightarrow`	$\upharpoonleft$	`\upharpoonleft`
$\twoheadleftarrow$	`\twoheadleftarrow`	$\twoheadrightarrow$	`\twoheadrightarrow`	$\upharpoonright$	`\upharpoonright`
$\leftarrowtail$	`\leftarrowtail`	$\rightarrowtail$	`\rightarrowtail`	$\downharpoonleft$	`\downharpoonleft`
$\leftrightharpoons$	`\leftrightharpoons`	$\rightleftharpoons$	`\rightleftharpoons`	$\downharpoonright$	`\downharpoonright`
$\Lsh$	`\Lsh`	$\Rsh$	`\Rsh`	$\rightsquigarrow$	`\rightsquigarrow`
$\looparrowleft$	`\looparrowleft`	$\looparrowright$	`\looparrowright`	$\leftrightsquigarrow$	`\leftrightsquigarrow`
$\curvearrowleft$	`\curvearrowleft`	$\curvearrowright$	`\curvearrowright`		
$\circlearrowleft$	`\circlearrowleft`	$\circlearrowright$	`\circlearrowright`		

Ponieważ w klasie `article` najwyższą jednostką w hierarchii podziału jest `\section` (czyli *punkt*), łatwo tworzy się książki (klasa `book`), w których rozdziałami są poszczególne artykuły. LaTeX dobierze za nas odpowiednie odstępy między rozdziałami oraz wielkość i krój pisma w śródtytułach.

Dwie z wymienionych instrukcji działają nieco inaczej niż pozostałe:

- Instrukcja `\part` nie ma wpływu na numerację rozdziałów.
- Instrukcja `\appendix` nie posiada argumentów. Jest to deklaracja zmieniająca z cyfr na litery sposób numerowania rozdziałów (w klasach `book` i `report`) lub punktów (w klasie `article`).

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

```
\tableofcontents
```

wstawia spis treści w miejscu, w którym jej użyjemy. Aby w spisie treści otrzymać prawidłowe numery stron, dokument trzeba przetworzyć („zlatechować”) dwukrotnie. Czasami niezbędna jest nawet trzecia kompilacja. Kiedy pod koniec przetwarzania dokumentu LaTeX wyświetli komunikat:

```
LaTeX Warning: Label(s) may have changed.
Rerun to get cross-references right.
```

oznacza to, że niezbędna jest kolejna kompilacja.

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 tyle proste, co skuteczne. Podczas przetwarzania dokumentu LaTeX zapisuje odpowiednie informacje do plików pomocniczych. Przeznaczenie danego pliku pomocniczego jest zasygnalizowane w rozszerzeniu nazwy. I tak: plik o rozszerzeniu `.toc` zawiera spis treści, plik `.lot` – spis tabel, plik `.lof` – spis rysunków, plik `.aux` – informacje o odsyłaczach wewnątrz dokumentu (odsyłacze omawiamy w podrozdziale 2.7). Informacje wprowadzone do plików pomocniczych LaTeX wykorzystuje podczas kolejnych przebiegów.

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 podrozdziału zostanie umieszczony w dokumencie, ale nie w spisie treści; tytuł nie jest też objęty numeracją. Przykładowo wersją „z gwiazdką” instrukcji `\section{Help}` jest `\section*{Help}`.

Najczęściej hasła w spisie treści są identyczne z odpowiadającymi im tytułami rozdziałów czy punktów. Czasami jednak nie jest to pożądane, ponieważ na przykład tekst hasła jest zbyt długi. W takich wypadkach hasło do spisu treści można podać jako *opcjonalny* argument instrukcji podziału hierarchicznego, np:

Tabela 3.5: Symbole operacji dwuargumentowych

$+$	`+`	$-$	`-`		
$\pm$	`\pm`	$\mp$	`\mp`	$\triangleleft$	`\triangleleft`
$\cdot$	`\cdot`	$\div$	`\div`	$\triangleright$	`\triangleright`
$\times$	`\times`	$\setminus$	`\setminus`	$\star$	`\star`
$\cup$	`\cup`	$\cap$	`\cap`	$\ast$	`\ast`
$\sqcup$	`\sqcup`	$\sqcap$	`\sqcap`	$\circ$	`\circ`
$\vee$	`\vee` , `\lor`	$\wedge$	`\wedge` , `\land`	$\bullet$	`\bullet`
$\oplus$	`\oplus`	$\ominus$	`\ominus`	$\diamond$	`\diamond`
$\odot$	`\odot`	$\oslash$	`\oslash`	$\uplus$	`\uplus`
$\otimes$	`\otimes`	$\bigcirc$	`\bigcirc`	$\amalg$	`\amalg`
$\bigtriangleup$	`\bigtriangleup`	$\bigtriangledown$	`\bigtriangledown`	$\dagger$	`\dagger`
$\lhd$	`\lhd` [a]	$\rhd$	`\rhd` [a]	$\ddagger$	`\ddagger`
$\unlhd$	`\unlhd` [a]	$\unrhd$	`\unrhd` [a]	$\wr$	`\wr`

Tabela 3.6: Symbole zmiennej wielkości

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

Tabela 3.7: Strzałki

$\leftarrow$	`\leftarrow` albo `\gets`	$\longleftarrow$	`\longleftarrow`	$\uparrow$	`\uparrow`
$\rightarrow$	`\rightarrow` albo `\to`	$\longrightarrow$	`\longrightarrow`	$\downarrow$	`\downarrow`
$\leftrightarrow$	`\leftrightarrow`	$\longleftrightarrow$	`\longleftrightarrow`	$\updownarrow$	`\updownarrow`
$\Leftarrow$	`\Leftarrow`	$\Longleftarrow$	`\Longleftarrow`	$\Uparrow$	`\Uparrow`
$\Rightarrow$	`\Rightarrow`	$\Longrightarrow$	`\Longrightarrow`	$\Downarrow$	`\Downarrow`
$\Leftrightarrow$	`\Leftrightarrow`	$\Longleftrightarrow$	`\Longleftrightarrow`	$\Updownarrow$	`\Updownarrow`
$\mapsto$	`\mapsto`	$\longmapsto$	`\longmapsto`	$\nearrow$	`\nearrow`
$\hookleftarrow$	`\hookleftarrow`	$\hookrightarrow$	`\hookrightarrow`	$\searrow$	`\searrow`
$\leftharpoonup$	`\leftharpoonup`	$\rightharpoonup$	`\rightharpoonup`	$\swarrow$	`\swarrow`
$\leftharpoondown$	`\leftharpoondown`	$\rightharpoondown$	`\rightharpoondown`	$\nwarrow$	`\nwarrow`
$\rightleftharpoons$	`\rightleftharpoons`	$\iff$	`\iff`	$\leadsto$	`\leadsto` [a]

[a] Dostępne po dołączeniu pakietu latexsym

Podobnie jak w przypadku spisów treści, tabel czy rysunków potrzebne są co najmniej dwie, a z reguły trzy, kompilacje dokumentu do prawidłowego wstawienia numerów odsyłaczy. Podczas pierwszej LaTeX wysyła do pliku pomocniczego (z rozszerzeniem `.aux`) informacje o odsyłaczach, które wykorzystuje podczas kolejnych kompilacji.

```
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 2.7 na stronie 30.”

2.8. Przypisy

Do składania przypisów u dołu strony służy instrukcja:

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

Instrukcję należy wstawić bezpośrednio po słowie lub zdaniu, do którego się odnosi[6].

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

```
Przypisy\footnote{To jest właśnie
przypis.} są często
wykorzystywane przez
użytkowników \LaTeX-a
```

Przypisy[a] są często wykorzystywane przez użytkowników LaTeX-a

[a]To jest właśnie przypis.

2.9. Wyróżnienia

W tekstach pisanych na maszynie fragmenty tekstu, które mają zostać wyróżnione, <u>podkreśla się</u>. W dokumentach drukowanych wyróżnianie fragmentów odbywa się przez składanie ich *kursywą*. Służy do tego instrukcja:

`\emph{`*text*`}`

Argumentem tej instrukcji jest tekst, który ma zostać wyróżniony.

```
\emph{\emph{Wyróżnienia} w~tekście
już wyróżnionym są składane
\emph{zwykłym} krojem pisma.}
```

Wyróżnienia *w tekście już wyróżnionym są składane* zwykłym *krojem pisma.*

[6]W krajach anglosaskich przypisy odnoszące się do całego zdania lub jego części umieszcza się natychmiast po kropce lub przecinku. W Polsce najczęściej umieszcza się je *przed* znakiem przestankowym (zasadę tę stosujemy w niniejszym wprowadzeniu).


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

$$\mu, M \quad \boldsymbol{\mu},\boldsymbol{M} \quad \pmb{\mu}, \pmb{M}$$

3.9. Zestawienie symboli matematycznych

W poniższych tabelach zestawiono wszystkie symbole standardowo dostępne w *trybie matematycznym*. Symbole w tabelach 3.12–3.16[10] są dostępne, jeżeli mamy zainstalowane dodatkowe fonty matematyczne (*AMS math fonts*) i do dokumentu dołączymy pakiet amssymb. Jeżeli brak jest fontów/pakietu to jest on dostępny w: `CTAN://tex-archive/macros/latex/packages/amslatex`.

Tabela 3.1: Akcenty matematyczne

$\hat{a}$	`\hat{a}`	$\check{a}$	`\check{a}`	$\tilde{a}$	`\tilde{a}`	$\acute{a}$	`\acute{a}`
$\grave{a}$	`\grave{a}`	$\dot{a}$	`\dot{a}`	$\ddot{a}$	`\ddot{a}`	$\breve{a}$	`\breve{a}`
$\bar{a}$	`\bar{a}`	$\vec{a}$	`\vec{a}`	$\widehat{A}$	`\widehat{A}`	$\widetilde{A}$	`\widetilde{A}`

Tabela 3.2: Małe litery alfabetu greckiego

α	`\alpha`	θ	`\theta`	o	`o`	υ	`\upsilon`
β	`\beta`	ϑ	`\vartheta`	π	`\pi`	ϕ	`\phi`
γ	`\gamma`	ι	`\iota`	ϖ	`\varpi`	φ	`\varphi`
δ	`\delta`	κ	`\kappa`	ρ	`\rho`	χ	`\chi`
ϵ	`\epsilon`	λ	`\lambda`	ϱ	`\varrho`	ψ	`\psi`
ε	`\varepsilon`	μ	`\mu`	σ	`\sigma`	ω	`\omega`
ζ	`\zeta`	ν	`\nu`	ς	`\varsigma`		
η	`\eta`	ξ	`\xi`	τ	`\tau`		

Tabela 3.3: Duże litery alfabetu greckiego

Γ	`\Gamma`	Λ	`\Lambda`	Σ	`\Sigma`	Ψ	`\Psi`
Δ	`\Delta`	Ξ	`\Xi`	Υ	`\Upsilon`	Ω	`\Omega`
Θ	`\Theta`	Π	`\Pi`	Φ	`\Phi`		

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

2.10.2. Środowiska `flushleft`, `flushright` i `center`

Środowiska `flushleft` i `flushright` składają akapity wyrównane, odpowiednio, do lewego lub prawego marginesu. Wewnątrz środowiska `center` każdy wiersz akapitu jest wyśrodkowany w osi szpalty. Tak jak zawsze, LaTeX dzieli akapity na wiersze automatycznie, w obrębie powyższych środowisk można jednak poleceniem `\\` wymusić zmianę wiersza.

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

To jest tekst
wyrównany do lewej. LaTeX nie składa tu
wierszy o jednakowej długości.

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

To jest tekst
wyrównany do prawej. LaTeX nie składa tu
wierszy
z zachowaniem jednakowej długości.

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

To jest tekst
wyśrodkowany.

2.10.3. Środowiska `quote`, `quotation` i `verse`

Środowisko `quote` nadaje się do składania cytatów oraz przykładów.

```
Jeżeli chodzi o~długości wiersza,
regułą kciuka jest, że:
\begin{quote}
Przeciętnie wiersz nie powinien
zawierać więcej niż 66 znaków.

Dlatego w~\LaTeX-u standardowe
strony mają szerokie marginesy.
\end{quote}
Dlatego też w~gazetach stosuje
się druk wieloszpaltowy.
```

Jeżeli chodzi o długości wiersza, regułą kciuka jest, że:

> Przeciętnie wiersz nie powinien zawierać więcej niż 66 znaków.
> Dlatego w LaTeX-u standardowe strony mają szerokie marginesy.

Dlatego też w gazetach stosuje się druk wieloszpaltowy.

Istnieją ponadto dwa środowiska o podobnym zastosowaniu: `quotation` oraz `verse`. Środowisko `quotation` jest przydatne do formatowania cytatów dłuższych niż jeden akapit. W przeciwieństwie do środowiska `quote`, wewnątrz środowiska `quotation` LaTeX rozpoczyna poszczególne akapity od wcięcia akapitowego. Środowisko `verse` służy do składania wierszy. Poszczególne linijki w zwrotkach należy kończyć instrukcją `\\` zaś poszczególne zwrotki oddzielać pustą linią.


```
\begin{displaymath}
\mathop{\mathrm{cov}}(X,Y)=
\frac{\displaystyle
\sum_{i=1}^n(x_i-\overline x)
(y_i-\overline y)}
{\displaystyle\biggl[
\sum_{i=1}^n(x_i-\overline x)^2
\sum_{i=1}^n(y_i-\overline y)^2
\biggr]^{1/2}}
\end{displaymath}
```

$$\mathrm{cov}(X,Y)=\frac{\displaystyle\sum_{i=1}^n(x_i-\overline{x})(y_i-\overline{y})}{\displaystyle\biggl[\sum_{i=1}^n(x_i-\overline{x})^2\sum_{i=1}^n(y_i-\overline{y})^2\biggr]^{1/2}}$$

Powyższy przykład ilustruje sytuację, w której należy wykorzystać polecenie `\bigg`, ponieważ nawiasy utworzone za pomocą konstrukcji `\left[` i `\right]` są zbyt małe.

3.7. Twierdzenia, definicje, itp.

W pracach matematycznych występuje potrzeba wyróżniania w składzie zapisu lematów, definicji, aksjomatów i tym podobnych elementów. Do tego celu mamy w LaTeX-u polecenia:

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

Argument *nazwa* oznacza nazwę środowiska. Argument *tekst* to nazwa elementu, która pojawi się na wydruku. Może to być „Twierdzenie”, „Definicja” itp. Argumenty w nawiasach kwadratowych są nieobowiązkowe. Za ich pomocą określamy sposób numerowania twierdzeń. Argument *nazwa'* to nazwa elementu, który zdefiniowano uprzednio poleceniem `\newtheorem`. Jeśli ten argument podano, to środowisko *nazwa* będzie numerowane w taki sam sposób co środowisko *nazwa'*. Ponadto oba środowiska mają wspólną a nie osobną numerację. Argument *punkt* dotyczy natomiast numerowania twierdzeń wewnątrz określonej jednostki podziału[9].

Po umieszczeniu instrukcji `\newtheorem{`*nazwa*`}...` w preambule, środowisko *nazwa* można wykorzystywać w dokumencie w następujący sposób:

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

Tyle teoria. Następujące przykłady, miejmy nadzieję, usuną wszelkie wątpliwości i jednocześnie uświadomią nam, że polecenie `\newtheorem` może być cokolwiek trudne do zrozumienia.

[9] Gdy umieścimy tam na przykład `chapter`, to elementy będą numerowane w obrębie rozdziałów.

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

```
gwiazdkowa␣wersja
środowiska␣verbatim
wyróżnia␣␣␣spacje
w␣tekście
```

Środowiska `verbatim` oraz instrukcji `\verb` nie wolno używać wewnątrz argumentów innych instrukcji.

2.10.5. Środowisko `tabular`

Do składania tabel służy środowisko `tabular`. LaTeX automatycznie ustala w nich szerokość poszczególnych rubryk. Środowisko ma jeden parametr obowiązkowy, określający liczbę kolumn tabeli i sposób ich justowania:

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

Dla każdej kolumny należy w argumencie wstawić jedną z liter: `l`, `r` lub `c`, określając w ten sposób justowanie zawartości kolumny. Dosunięcie zawartości kolumny do lewej oznaczamy literą `l`, do prawej – znakiem `r`, a wyśrodkowanie – znakiem `c`. Zapisu `p{`*szer-kolumn*`}` można użyć do zaznaczenia, że kolumna ma mieć szerokość *szer-kolumn*. Wewnątrz takiej kolumny tekst jest składany w prostokąt o zadanej szerokości, z wyrównywaniem obu marginesów. Znak `|` oznacza, że kolumny mają zostać rozdzielone pionową kreską.

Wewnątrz środowiska `tabular` poszczególne wiersze oddzielamy instrukcją `\\`, a rubryki w wierszu znakiem `&`. Instrukcja `\hline` wstawia poziomą linię na całą szerokość tabeli.

```
\begin{tabular}{|r|l|} \hline
7C0 & heksadecymalnie \\
3700 & oktalnie \\
11111000000 & binarnie \\
\hline \hline
1984 & dziesiętnie \\ \hline
\end{tabular}
```

7C0	heksadecymalnie
3700	oktalnie
11111000000	binarnie
1984	dziesiętnie

```
\begin{tabular}{|p{4.7cm}|} \hline
Ten akapit jest wewnątrz pudełka.
Mamy nadzieję, że uzyskany
efekt się podoba.\\ \hline
\end{tabular}
```

Ten akapit jest wewnątrz pudełka. Mamy nadzieję, że uzyskany efekt się podoba.

Instrukcją `@{...}` określamy odstęp między kolumnami. Instrukcja ta usuwa domyślny odstęp międzykolumnowy, zastępując go tym, jaki wyspecyfikowano wewnątrz `{...}`. Stosuje się ją często do wyrównywania zestawień


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

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

Środowiskiem `array` możemy się posłużyć także do składania wyrażeń zawierających tylko jeden ogranicznik, po prawej lub lewej stronie, stosując konstrukcję `\right.` (lub `\left.`).

```
\begin{displaymath}
y = \left\{ \begin{array}{ll}
a & \textrm{jeżeli $d>c$}\\
b+x & \textrm{rano}\\
l & \textrm{w~ciągu dnia}
\end{array} \right.
\end{displaymath}
```

$$y = \left\{ \begin{array}{ll} a & \textrm{jeżeli } d>c \\ b+x & \textrm{rano} \\ l & \textrm{w ciągu dnia} \end{array} \right.$$

Do składu wyrażeń wielowierszowych można zamiast środowiska `equation` użyć środowisk `eqnarray` lub `eqnarray*`. W środowisku `eqnarray` każdy wiersz zawartego w nim wyrażenia posiada osobny numer; w środowisku `eqnarray*` wiersze nie są numerowane. Działanie środowisk `eqnarray` oraz `eqnarray*` jest zbliżone do trzykolumnowej tabeli typu `{rcl}`. W takiej tabeli w środkowej kolumnie wstawiamy zwykle znaki równości lub nierówności. Poleceniem `\\` łamiemy poszczególne wiersze tej tabeli (środowiska).

```
\begin{eqnarray}
f(x) & = & \cos x \\
f’(x) & = & -\sin x \\
\int_{0}^{x} f(y)dy &
= & \sin x
\end{eqnarray}
```

$$f(x) = \cos x \tag{3.5}$$
$$f'(x) = -\sin x \tag{3.6}$$
$$\int_0^x f(y)dy = \sin x \tag{3.7}$$

Zwróćmy uwagę, że po obu stronach znaku równości odstęp, który wstawił tam LaTeX, jest zbyt duży. Możemy go zmniejszyć za pomocą polecenia `\setlength\arraycolsep{2pt}`.

LaTeX nie dzieli automatycznie wyrażeń nie mieszczących sie w jednym wierszu. Musimy to zrobić sami. Najczęściej w taki oto sposób:

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

$$\begin{aligned} \sin x = {} & x - \frac{x^3}{3!} + \frac{x^5}{5!} - \\ & - \frac{x^7}{7!} + \cdots \end{aligned} \tag{3.8}$$

Tabela 2.2: Opcjonalny argument środowisk `table` i `figure`

Znak	Dopuszczalne miejsce umieszczenia wstawki
`h`	bez przemieszczenia, dokładnie w miejscu użycia (użyteczne w odniesieniu do niewielkich wstawek);
`t`	na górze strony;
`b`	na dole strony;
`p`	na stronie zawierającej wyłącznie wstawki;
`!`	ignorując większość parametrów kontrolujących umieszczanie wstawek[a], przekroczenie wartości, których może nie pozwolić na umieszczanie następnych wstawek na stronie.

[a]Są to takie parametry, jak na przykład maksymalna dopuszczalna liczba wstawek na stronie.

Do tworzenia wstawek mamy w LaTeX-u dwa środowiska. Środowisko `figure` służy do tworzenia rysunków, a środowisko `table` – do tablic. Oba mają jeden parametr opcjonalny:

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

Argument *miejsce* określa, gdzie na stronie można umieścić wstawkę. Jest to sekwencja od jednego do pięciu znaków: `h`, `t`, `b`, `p` oraz `!`. Każdy znak określa dopuszczalny sposób umieszczenia wstawki; szczegółowe informacje zestawiono w tabeli 2.2.

Przykładowa tabela może zaczynać się tak:

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

Argument `[!hbp]` oznacza, że tabelę można umieścić w miejscu, w którym pojawia się w pliku źródłowym (`h`) albo na dole strony (`b`), albo wreszcie na osobnej stronie zawierającej wyłącznie wstawki (`p`). Ponadto wstawienie „`!`” oznacza, że LaTeX ma pominąć większość parametrów kontrolujących umieszczanie wstawek. Jeżeli instrukcji `\table` użyto bez opcjonalnego argumentu, to jego domyślnymi wartościami są `[tbp]`.

LaTeX umieszcza każdą wstawkę zgodnie ze specyfikacją autora. Jeżeli wstawki nie można umieścić na bieżącej stronie, to dołącza się ją albo do *kolejki rysunków*, albo do *kolejki tabel*[8]. Po rozpoczęciu składania nowej strony LaTeX sprawdza, czy można zapełnić całą stronę wstawkami czekającymi w kolejce; gdy nie jest to możliwe, pierwszą wstawkę z każdej kolejki traktuje w taki sposób, jak gdyby właśnie pojawiła się w tekście: LaTeX stara się ją umieścić zgodnie z wartościami parametru *miejsce* (za wyjątkiem `h`, gdyż nie jest to już oczywiście możliwe). Nowe wstawki są odpowiednio dołączane do kolejek. LaTeX dba o właściwy porządek wstawek każdego typu.

[8]Są to kolejki typu FIFO (pierwsze weszło – pierwsze wyszło).

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 `\left` oraz ogranicznika wymaga nawiasu zamykającego poprzedzonego poleceniem `\right`. Gdy ogranicznik ma się pojawić tylko po jednej stronie, wówczas po drugiej *należy* użyć konstrukcji `\left.` (po lewej) lub `\right.` (po prawej).

```
\begin{displaymath}
1 + \left( \frac{1}{ 1-x^{2} }
\right) ^3
\end{displaymath}
```

$$1+\left(\frac{1}{1-x^2}\right)^3$$

W pewnych sytuacjach należy samemu określić właściwą wielkość ogranicznika. Do tego celu służą instrukcje `\big`, `\Big`, `\bigg` oraz `\Bigg`, poprzedzające odpowiedni ogranicznik[4].

```
$\Big( (x+1) (x-1) \Big) ^{2}$\\
$\big(\Big(\bigg(\Bigg($\quad
$\big\}\Big\}\bigg\}\Bigg\}$\quad
$\big\|\Big\|\bigg\|\Bigg\|$
```

$\Big((x+1) (x-1) \Big) ^{2}$

$\big(\Big(\bigg(\Bigg($ $\quad$ $\big\}\Big\}\bigg\}\Bigg\}$ $\quad$ $\big\|\Big\|\bigg\|\Bigg\|$

Wielokropek w wyrażeniach matematycznych wprowadzamy poleceniem `\ldots`. Kropki pojawiają się wtedy na linii podstawowej, to znaczy na jednakowej wysokości z przecinkiem czy kropką. Instrukcja `\cdots` wstawia natomiast inny rodzaj wielokropka, w którym kropki znajdują się w osi znaków $+$, $-$, $=$. Ponadto są jeszcze instrukcje `\vdots` oraz `\ddots`. Pierwszą z nich składamy wielokropki pionowe, a drugą skośne (zobacz przykład w punkcie 3.5).

```
\begin{displaymath}
x_{1},\ldots,x_{n} \qquad
x_{1}+\cdots+x_{n}
\end{displaymath}
```

$$x_{1},\ldots,x_{n} \qquad x_{1}+\cdots+x_{n}$$

3.4. Odstępy w trybie matematycznym

Zdarzają się sytuacje, kiedy wielkość odstępów wewnątrz wyrażeń matematycznych jest nieodpowiednia. Można jednak skorygować je samemu za pomocą odpowiednich instrukcji. Do wprowadzenia niewielkich odstępów służy kilka poleceń: `\,` wstawia odstęp równy $\frac{3}{18}$ kwadratu (⊔), `\:` pozwala uzyskać

[4]Polecenia te nie działają zgodnie z oczekiwaniami, jeżeli uprzednio wykorzystano instrukcje zmieniające stopień pisma albo opcje `11pt` lub `12pt`. W takiej sytuacji należy skorzystać z pakietu `exscale` albo pakietu `amsmath`.

W powyższym przykładzie[9] LaTeX najpierw spróbuje umieścić rysunek bez przesuwania go dokądkolwiek (h). Jeżeli okaże się to niemożliwe, spróbuje umieścić go na dole strony (b). Jeżeli i to okaże się niewykonalne, będzie się starał umieścić rysunek na stronie zawierającej wyłącznie wstawki (p). Jeżeli w kolejkach rysunków i tabel nie ma wstawek pozwalających wypełnić stronę LaTeX rozpocznie nową stronę i spróbuje umieścić na niej rysunek traktując go znowu tak, jakby właśnie pojawił się w tekście.

Czasami może okazać się niezbędne wykonanie 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 przypadku użycia instrukcji `\cleardoublepage` LaTeX rozpoczyna od stronie nieparzystej.

W dalszej części tego wprowadzenia przedstawimy, jak można dołączać do LaTeX-owego dokumentu rysunki w formacie PostScript (por. 4.1).

2.12. Listy

Do pisania listów można wykorzystać klasę `letter`. Struktura pliku źródłowego w przypadku listów oczywiście różni się od dokumentów z klasy `article` czy `book`. Klasę `letter` zaprojektowano tak, że bezproblemowo możemy napisać pojedynczy list do przysłowiowej „cioci”, jak i setki listów do różnych osób (korespondencja seryjna).

Jeśli imię, nazwisko i adres nadawcy mają być identyczne we wszystkich listach, to deklarujemy je za pomocą polecenia (instrukcja `\\` oznacza złamanie wiersza):

`\address{`*imię*`\\`*nazwisko*`\\` *adres* ... `}`

Polecenie `\signature` służy do zadeklarowania podpisu pod listem. Wewnątrz argumentu polecenia `\signature` instrukcja `\\` służy do rozpoczęcia nowego wiersza, przykładowo:

```
Dyr. E.~K.~Tor,\\ Przewodniczący
Zastępcy
```

Dyr. E. K. Tor,
Przewodniczący Zastępcy

Data jest wstawiana automatycznie. Jeżeli chcemy wstawić inną, należy zastosować deklarację `\date`:

`\date{16 Czerwca 1963~r.}`

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

Treść listu powinniśmy wpisać wewnątrz środowiska `letter`. Środowisko to ma jeden argument, którym jest adres osoby, do której piszemy. Wewnątrz środowiska `letter` można skorzystać z kilku prostych poleceń służących do umieszczania w odpowiednim

[9] Zakładając, że kolejka rysunków jest pusta.

umożliwia nam instrukcja `\surd`[3]. Natomiast pierwiastek stopnia n składamy konstrukcją `\sqrt[`n`]`.

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

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

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

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

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

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

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

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

Akcenty matematyczne, takie jak daszki czy tyldy nad zmiennymi, umieszczamy we wzorze za pomocą instrukcji zestawionych w tabeli 3.1. Szerokie daszki i tyldy, obejmujące wiele symboli, wstawiamy używając instrukcji `\widetilde` oraz `\widehat`. Użycie znaku „`'`” powoduje wstawienie symbolu „prim”.

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

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

Niekiedy **wektory** wyróżniamy, wstawiając akcent 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:

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

[3] Taki zapis jest wykorzystywany raczej w literaturze anglosaskiej.

Rozdział 3

Wyrażenia matematyczne

Nareszcie! W tym rozdziale poznasz najlepszą stronę TEX-a, czyli skład wzorów matematycznych. Ostrzegamy jednak, że przedstawimy tu tylko absolutne podstawy. Chociaż większości użytkowników one wystarczają, nie załamuj rąk, jeżeli nie poradzisz sobie z jakimś skomplikowanym wzorem, tylko raczej zapoznaj się z możliwościami pakietu AMS-LATEX[1] lub innego specjalizowanego pakietu.

3.1. Wstęp

Do składu wyrażeń matematycznych mamy w LATEX-u specjalny *tryb matematyczny*. Oznacza to wpisywanie wzorów pomiędzy `\(` i `\)` albo `$` i `$`, albo pomiędzy `\begin{math}` oraz `\end{math}`.

```
$a$ do kwadratu plus~$b$
do kwadratu równa się~$c$
do kwadratu. Albo, stosując
bardziej matematyczne
podejście, $c^{2}=a^{2}+b^{2}$.
```

a do kwadratu plus b do kwadratu równa się c do kwadratu. Albo, stosując bardziej matematyczne podejście, $c^2 = a^2 + b^2$.

```
{\TeX} należy wymawiać jako
$\tau\epsilon\chi$.\\[6pt]
100~m$^{3}$ wody \\[6pt]
To płynie z~mojego~$\heartsuit$
```

TEX należy wymawiać jako $\tau\epsilon\chi$.

100 m^3 wody

To płynie z mojego $\heartsuit$

Składając większe wzory, powinniśmy je eksponować, to znaczy wstawiać między akapitami w osobnym wierszu. Takie wzory umieszcza się albo pomiędzy `\[` i `\]`, albo wewnątrz środowiska `displaymath`. Ta ostatnia konstrukcja dotyczy tworzenia wzorów bez żadnej numeracji. Do automatycznego numerowania wzorów stosujemy natomiast środowisko `equation`.

[1] `CTAN://tex-archive/macros/latex/packages/amslatex`.


```
$a$ 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 w~drugim akapicie.
```

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

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

Pierwszy wiersz w drugim akapicie.

Instrukcje `\label` oraz `\ref` służą do tworzenia odsyłaczy do równań.

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

$$\epsilon > 0 \tag{3.1}$$

Ze wzoru (3.1) otrzymujemy ...

Zwróćmy uwagę, że inaczej składa się wzory wewnątrz akapitu, a inaczej eksponowane:

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

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

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

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

Między *trybem matematycznym* LATEX-a a *trybem tekstowym* istnieją duże różnice. Na przykład w *trybie matematycznym*:

1. LATEX ignoruje prawie wszystkie odstępy oraz znaki końca linii; wszystkie odstępy we wzorach wynikają albo z kontekstu albo z użycia specjalnych poleceń, takich jak `\,`, `\quad` lub `\qquad`.
2. Puste linie są niedozwolone. Obowiązuje zasada: jeden wzór – jeden akapit.
3. Litery we wzorach służą do oznaczania nazw zmiennych; zmienne składamy inaczej niż zwykły tekst. Jeżeli częścią wzoru jest zwykły tekst, to należy posłużyć się instrukcją `\textrm{...}`.

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

$$\forall x \in \mathbf{R}\colon \qquad x^{2} \geqslant 0 \tag{3.2}$$

```
\begin{equation}
x^{2} \geq 0\qquad
\textrm{dla każdego }x\in\mathbf{R}
\end{equation}
```

$$x^{2} \geqslant 0 \qquad \textrm{dla każdego } x \in \mathbf{R} \tag{3.3}$$