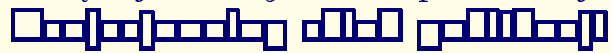


KAZIMIERZ M. BORKOWSKI



Profesjonalny skład publikacji



Toruń, 1992



Spis treści

1. Czym jest \LaTeX ?	3
2. Pierwsze kroki	5
<i>Przykład publikacji</i>	6
3. Ogólna redakcja pliku źródłowego	7
<i>Style i opcje dokumentu</i>	7
<i>Style stron</i>	8
<i>Strona tytułowa</i>	9
<i>Podział na sekcje</i>	9
<i>Spisy treści, tabel i rysunków</i>	10
4. Ważne subtelnosci	10
5. Jednostki \TeX-a	11
6. Rozmiary i kroje czcionek (fonty)	11
<i>Rozmiary, rodziny i opcje</i>	11
<i>Tabela znaków antykwy i symboli math</i>	13
<i>Rozdzielczość i powiększenie</i>	14
7. Strona, paragraf	15
<i>Parametry stylów</i>	17
8. Bloki i ramki (<i>boxes</i>)	18
9. Linie	20
10. Otoczenia (<i>environments</i>)	21
11. Tabele i rysunki	22
12. Etykiety, odwołania i przypisy	26
13. Liczniki (<i>counters</i>)	27
14. Znaki diakrytyczne (akcenty)	28
15. Wyrażenia matematyczne	29
<i>Tabele poleceń trybu matematycznego</i>	32
16. Definicje i makrodefinicje	35
17. Kilka przykładów makrodefinicji	38
<i>Inicjał akapitowy</i>	38
<i>Konstruowanie obramowań</i>	39
<i>Dynamiczne odstępy</i>	40

<i>Pętle</i>	41
<i>Definicja „polskich” znaków</i>	42
18. Jeszcze kilka użytecznych poleceń	43
19. T_EX — L^AT_EX	44
<i>Lista poleceń PLAIN niedostępnych w L^AT_EX-u</i>	44
<i>Przygotowanie pliku formatu</i>	44
20. Podział pliku źródłowego	45
21. Formowanie pliku .dvi	45
22. Jeszcze o fontach	47
22.1 Rodziny	47
22.2 Parametry fontów	47
<i>Tabela fontów T_EX-a</i>	48
22.3 Bezpośredni dostęp do fontów	49
<i>Tabele znaków wybranych fontów</i>	51
23. Źródła	55
<i>Literatura</i>	55

1. Czym jest L^AT_EX?

Jest to obszerny pakiet makrodefinicji — poleceń wyższego poziomu — znacznie upraszczający korzystanie z wysoce wyrafinowanego programu do profesjonalnego składu tekstów, zwanego T_EX-em¹. Jest to jednocześnie rozszerzenie i tak już nadzwyczaj bogatych możliwości T_EX-a, gdyż każde z jego prymitywnych poleceń jest dostępne w L^AT_EX-u.

Korzystanie z T_EX-a, bądź L^AT_EX-a, polega na dostarczeniu temu systemowi pewnych słów, równań, rysunków, tabel itd., aby otrzymać plik zawierający skład gotowy do wydruku. Jeśli dysponujemy urządzeniem drukującym o odpowiednio wysokiej rozdzielczości, możemy ten plik przenieść na papier i wręczyć wydawcy, który z kolei przeniesie go bezpośrednio na płyty w celu powielenia w tysiącach egzemplarzy.

Ten uproszczony opis może sugerować, że L^AT_EX jest rodzajem super-edytora, lecz taka analogia jest niefortunna i to z kilku powodów. Przede wszystkim T_EX w czasie swej pracy zajmuje się szeregiem typograficznych niuansów, które są całkowicie ignorowane (faktycznie są poza ich zasięgiem) przez tradycyjne edytory czy programy przetwarzania tekstów. System taki niektórzy nazywają *paginatorem*. Wśród wspomnianych niuansów wyróżnia się „kerning” i ligatury. Kerning polega na dodaniu lub odjęciu specjalnego odstępu od normalnego odstępu między literami w słowie w celu poprawy wyglądu tekstu. Wartość poprawki zależy od sąsiadujących liter. Porównajmy np.

WATA — skład z kerningiem i

WATA — to samo bez kerningu.

Ligatury zaś, to specjalne znaki reprezentujące dwa lub więcej zwykłych znaków. W piśmie rzymskim są to np. fi i fl używane w miejsce fi i fl.

Poza T_EX-em chyba nie ma innego systemu, który tak świetnie składa nawet bardzo skomplikowane formuły matematyczne.

Innym wyróżnikiem T_EX-a spośród edytorów tekstu jest „cykl życiowy”. Zwykły edytor pracuje dwustopniowo:

- wprowadzenie tekstu
- wydrukowanie go.

„T_EX-owanie” wymaga kilku etapów:

- (1) przygotowanie pliku źródłowego
- (2) przetworzenie go T_EX-em
- (3) powrót do kroku (1), jeśli T_EX wykryje błędy
- (4) przejrzanie wyniku przetwarzania na monitorze
- (5) wydruk na drukarce, albo powrót do kroku (1) w celu wprowadzenia poprawek.

W związku z tym, kompletny system T_EX-a obejmuje kilka różnych współpracujących

¹Słowo T_EX wymawia się jak „tech”. Jest to zgodne z sugestią autora systemu, który wywodzi tą nazwę z greckiego rdzenia $\tau\epsilon\chi$... oznaczającego zarówno sztukę, jak i technologię. Opuszczona litera E w nazwie symbolizuje sztukę drukarską i odróżnia ją od podobnych nazw innych systemów. Gdy nie mamy możliwości opuszczenia E, to proponuje się pisać TeX. Zatem L^AT_EX będziemy czytali tak, jak „latech”, „lejtech”, „latek” albo „lejtec”

programów: dowolny edytor znaków ASCII [w kroku (1)], program główny [krok (2)], „previewer” — do przeglądania wyników przetwarzania [krok (4)], program tłumaczący tenże wynik na kod zrozumiały dla danego urządzenia drukującego [krok (5)] i, ewentualnie, program nadzorujący ułatwiający użytkownikowi wybór kroków i wywoływanie programów systemu (poprzez menu).

Niewątpliwie więc praca z \TeX -em jest bardziej złożona niż np. z ChiWriter-em czy WordStar-em. Przyczyna tkwi w samym sednie składu tekstowego. Istnieje wszak nieskończona ilość formatów, spośród których chcemy mieć możliwość wyboru przy przygotowywaniu dokumentów czy publikacji o profesjonalnej jakości. Przy ustalonym formacie, umiejscowienie jakiegoś elementu typograficznego na stronie na ogół zależy tak od materiału poprzedzającego go jak i po nim następującego. Nie jest wcale rzadkością, że poprawne złamanie pierwszego wiersza akapitu zależy od ostatniego słowa w akapicie! W przekonaniu wytrawnych użytkowników \TeX -a żaden inny system komputerowego składu tekstów nie dorównuje jakością wyniku i zakresem możliwości formatowania stron \TeX -owi. Mówi się, że kto raz zasmakował owoców \TeX -a, ten już wkrótce zapomni jak kiedyś cenił edytory typu ChiWriter (autor niniejszego tekstu doświadczył tego na sobie).

Trzeba wyjaśnić, że w praktyce rzadko kto korzysta bezpośrednio z prymitywnych rozkazów \TeX -a. Kiedy słyszymy, że tekst został złożony \TeX -em, to na ogół trafne będzie przypuszczenie, iż używano tzw. formatu PLAIN, tj. typowego zbioru definicji poleceń wyższego rzędu dołączanego normalnie do kompletu programów, zbiorów źródłowych i zestawów czcionek składających się na system \TeX . Drugą przypuszczalną możliwością jest \LaTeX . W istocie \LaTeX jest dalszym rozszerzeniem PLAIN-a i, z kilkudziesięciu zaledwie wyjątkami, obejmuje wszystkie polecenia PLAIN (a jest ich blisko 1000!). Ograniczenie się do poleceń samego \LaTeX -a zubaża znacznie przebogata ofertę \TeX -a, ale jednocześnie uwalnia nas od opanowywania wielu setek prymitywniejszych poleceń. I chociaż niewątpliwie zadowolą nas wyniki początkowych eksperymentów, to w miarę nabywania doświadczenia i praktyki z pewnością zechcemy nie jeden raz sięgnąć do tego źródła. Z tego powodu w niniejszym opracowaniu podajemy obok wszystkich najważniejszych \LaTeX -owych wiele z poleceń PLAIN-a i samego \TeX -a. W żadnym wypadku nie można jednak traktować prezentowanego tekstu jako podstawowego materiału o paginatorze \TeX . Spośród dziesiątków pozycji literaturowych najważniejszymi i podstawowymi są książki Profesora Donalda E. Knutha ze Stanford University — twórcy \TeX -a. Podobnie, jedynie *The \LaTeX Document Preparation System* Leslie Lamporta jest kompletnym i autorytatywnym źródłem wiedzy o \LaTeX -u. Niedawno pojawiło się polskie tłumaczenie (P. Wyrostka) tej ważnej pozycji z tytułem *\LaTeX* .

\TeX zyskał na świecie dużą popularność. W Stanach Zjednoczonych jest on standardem **American Mathematical Society**. Coraz więcej znanych czasopism naukowych akceptuje dyskietkowe manuskrypty artykułów napisanych w \TeX -u (np. *Astronomy and Astrophysics*, *Celestial Mechanics*) — także w Polsce (*Acta Astronomica*, *Postępy Fizyki*). W USA powstała organizacja użytkowników \TeX -a (TUG, od \TeX Users Group) wydająca własne czasopismo i gromadząca kilka tysięcy indywidualnych członków i około 150 instytucji z całego świata. Jak się dowiadujemy, w Polsce właśnie powstaje krajowy odpowiednik TUG — Grupa Użytkowników Systemu \TeX -a (GUST). Tymczasowy zarząd jest osiągalny pod adresem:

IBS PAN, ul. Newelska 6, 01-447 Warszawa

E-mail: GUST@PLCAMK61 lub GUST@ALFA.CAMK.EDU.PL

Niechaj Czytelnik nie przeoczy faktu, że system \TeX należy do oprogramowania typu *public domain*, co znaczy, że jest w zasadzie osiągalny po kosztach przesłania i nośnika (dyskiety czy taśmy magnetycznej). W praktyce, za pewne implementacje trzeba jednak płacić. Istotne jest również to, że istnieje wiele wersji tego paginatora dla różnych konfiguracji (w tym na komputery osobiste typu IBM PC i komputery „prawdziwe”). Jeśli nie znajdziemy wersji na jakiś specyficzny komputer, to zawsze istnieje możliwość skompilowania całkiem na nowo programu wykonawczego \TeX z programów źródłowych dostępnych w Pascalu i w języku C (albo w języku WEB, który jest na wyższym poziomie od Pascala i pozwala uzyskać nową wersję paskalową, odpowiednią do danego komputera).

Ze względu na liczebność, trudno byłoby przedstawić wszystkie wersje systemów opartych na \TeX -u. Wspomnimy tylko, że istnieją systemy do specyficznych języków i alfabetów (np. \jTeX dla japońskiego, \sbTeX-xet dla hebrajskiego, cyrylica), w tym kilka wersji do różnojęzycznych alfabetów łacińskich (np. \sbTeX , \emTeX czy \TÊX , tj. \TeX wielojęzyczny). Opracowano także szereg systemów rozszerzenia \LaTeX -a: \BIBTeX — do korzystania z bibliotek literatury, \SLITeX — do przygotowania przezroczy (slajdów), \PICTeX — do robienia rysunków, \IdxTeX i \GloTeX — do zestawiania spisów haseł i słowników w większych publikacjach książkowych. Istnieje też polska wersja samego \LaTeX -a (\LALeX) opracowany przez Polski \TeX ².

Niniejsze opracowanie kierujemy głównie do użytkowników komputerów osobistych kompatybilnych z IBM PC. Wersje \TeX -a dla tych komputerów noszą nazwy \emTeX , \sbTeX , \PCTeX i in. Sposób przygotowania samego pliku źródłowego dla \TeX -a czy \LaTeX -a nie będzie jednak istotnie zależał od konfiguracji ani typu komputera. Mamy też nadzieję, że Czytelnik nie będzie zmuszony do samodzielnego instalowania \TeX -a na swoim komputerze, a jeśli taka potrzeba zaistnieje to zwróci się on do lokalnego *guru* czy, jak niektórzy mówią, \TeX perta, gdyż objaśnienie tych spraw wykracza daleko poza ramy niniejszego opracowania.

2. Pierwsze kroki

Z niewielką pomocą \TeX perta uzyskanie pierwszych prostych wydruków z użyciem paginatora \LaTeX okaże się niewątpliwie sprawą niemal trywialną. Dalsze postępy osiągniemy szybko przeglądając zbiory `small.tex` i `sample.tex` (zwykle w katalogu `\texinput` albo `\inputs`). Możemy je ewentualnie zmodyfikować (oczywiście po skopiowaniu na swój katalog!) poznając skutki wprowadzonych zmian na wydruku. Jeśli Czytelnik jednak nie zna angielskiego, to chwilowo musi oprzeć się na naszym materiale, a jest on na tyle bogaty, że powinien wystarczyć na niezłe opanowanie tej sztuki nawet bez korzystania z pełniejszych opracowań.

Tekst, który chcemy pięknie wydrukować, wpisujemy najzwyczajniej do komputera lub na dyskietkę unikając jednak edytorów, które wstawiają specjalne znaki kontrolne, spoza zestawu znaków ASCII. W pliku tak utworzonym wystąpi tylko kilka instrukcji dla \TeX -a mówiących m.in. o rodzaju dokumentu i miejscu jego zakończenia. Instrukcje takie będą słowami zaczynającymi się od odwróconej kreski pochyłej, tj. od „backslasha”: `\`. Znak ten należy do kilku wyjątków o specjalnym znaczeniu. Prawie wszystkie polecenia \TeX -a rozpoczynają się właśnie od niego i kończą znakiem różnym od litery. Poza zna-

²Z Sopotu. Obecnie oddział firmy MacroSoft, Warszawa

kiem \ specjalne znaczenie mają także:

{ } % \$ # & -

Nawiasy klamrowe obejmują argumenty rzeczywiste wielu poleceń. Służą one także do wydzielenia obszaru tekstu, którego mają dotyczyć umieszczone tam polecenia. W pliku źródłowym wstawiamy znak % w celu wykasowania tekstu od tego miejsca do końca wiersza (fragment taki jest ignorowany przez T_EX-a). Jest to wygodny sposób wpisywania komentarzy do pliku źródłowego. Znak \$ otwiera i zamyka tekst traktowany jako wzór matematyczny. Między takimi znakami T_EX znajduje się w trybie *math* (matematycznym). Znak # używamy do zaznaczania argumentów formalnych w definicjach poleceń, zaś & wskazuje miejsce rozgraniczające kolumny w tabelach. Wreszcie znak - służy do zaznaczania dolnych wskaźników (subskryptów) w trybie matematycznym. W zwykłym tekście wszystkie wymienione symbole możemy zamieścić poprzedzając je znakiem \ (np. \{).

Na następnych dwóch stronach podajemy prosty przykład pełnego pliku źródłowego i praktycznie dosłowny wynik jego przetworzenia przez system L^AT_EX.

Warto zwrócić uwagę na całkiem poprawne łamanie polskich wyrazów bez dodatkowej pomocy użytkownika. Dzieje się tak dlatego, że T_EX wcześniej wczytał specjalną tablicę wzorów podziału na sylaby — różną dla każdego języka. Dla polskiej wersji językowej taką tabelę opracowała Hanna Kołodziejaska z Instytutu Informatyki UW (obecnie w firmie Microsoft; używamy jej w tej pracy z nieznacznymi zmianami).

Jeśli w tekście wpiszemy dowolnie długi ciąg odstępów (*spacji*), to T_EX zignoruje wszystkie prócz jednego. Pojedyncza zmiana wiersza traktowana jest również jak odstęp (T_EX sam wybiera miejsca złamania wiersza tekstu), ale podwójna, tzn. pusta linia, jest traktowana jako koniec ustępu. Ustęp zaczyna się zwykle od wcięcia i jest normalnie nazywany akapitem. My, ze względów T_EXnicznych, będziemy taki ustęp nazywali *paragrafem* (termin ten formalnie odnosi się do ustępu lub ustępów poprzedzonych znakiem paragrafu: § albo ¶). Dalsze puste wiersze poza pierwszym są ignorowane. Nowy wiersz nie zaczynający paragrafu możemy wymusić pisząc \.

```
\documentstyle{article}           % pocz/atek
\textwidth10.5cm                 % szeroko/s/c kolumny
\begin{document}                 % tu zaczyna si/e tekst
```

```
\centerline{\Large\bf Przyk/lad dokumentu}
\vspace{5mm}
S/lowa i zdania zwyk/lego tekstu w pliku
/xr/od/lowym \LaTeX-a oddziela si/e odst/epami.
Wi/ecej kolejnych odst/ep/ow          jest
r/ownowa/zne jednemu. R/ownie/z zmiany wiersza
traktowane s/a tak, jak odst/epy.
```

Jeden, lub wi/ecej (nast/epne s/a ignorowane),
 pustych wierszy oznaczaj/a koniec akapitu.
 Znak \% stawiamy na pocz/atku komentarza.
 Polecenia systemu zaczynaj/a si/e od odwrotnej

pochy/lej kreski, a ko/ncz/a znakiem r/o/znym od `{\it litery}`.

Miejsca dzielenia wyraz/ow, kt/orych `\TeX` nie rozpoznaje automatycznie mo/zemy wskaza/c poleceniem `\backslash$--`, np. `\{\tt jesz$\backslash$-cze}`.

```
\hfill A.U. Tor\
\end{document} Tak ko/nczy si/e plik zr/od/lowy;
cokolwiek jest dalej, TeX ignoruje
```

Przykład dokumentu

Słowa i zdania zwykłego tekstu w pliku źródłowym \LaTeX -a oddziela się odstępami. Więcej kolejnych odstępów jest równoważne jednemu. Również zmiany wiersza traktowane są tak, jak odstępy.

Jeden lub więcej (następne są ignorowane) pustych wierszy oznaczają koniec akapitu. Znak `%` stawiamy na początku komentarza. Polecenia systemu zaczynają się od odwrotnej pochyłej kreski, a kończą znakiem różnym od *litera*.

Miejsca dzielenia wyrazów, których \TeX nie rozpoznaje automatycznie możemy wskazać poleceniem `\-`, np. `jesz\ -cze`.

A.U. Tor

3. Ogólna redakcja pliku źródłowego

Zbiór źródłowy dla \LaTeX -a, nazwijmy go tutaj roboczo `plik.tex`, rozpoczyna się od komendy

```
\documentstyle[opcje]{styl},
```

gdzie `[opcje]` możemy całkowicie opuścić albo ujmując w nawiasy kwadratowe wpisać jedną (lub kilka oddzielonych przecinkami) z następujących możliwości:

<code>11pt</code>	<code>12pt</code>	<code>twoside</code>
<code>twocolumn</code>	<code>proc</code>	<code>titlepage</code>
<code>bezier</code>	<code>fleqn</code>	<code>leqno</code>
<code>ifthen</code>	<code>openbib</code>	

oznaczających rozmiar podstawowej czcionki (normalnie 10 pt, tj. punktów), rozróżnienie między formatami stron nieparzystych i parzystych (`twoside`), dwukolumnowe strony, format tzw. *proceedings* (materiałów z konferencji), stronę tytułową (te dwie używane ze stylem `article`), pewne dodatkowe zbiory poleceń do rysowania, formatowania wyrażeń matematycznych (`bezier`, `fleqn`, `leqno`), kontroli czytania pliku źródłowego (`ifthen`) i wspomaganie redakcji odniesień literaturowych z użyciem bibliograficznej bazy danych. Dla formatów dwukolumnowych możemy zamiast opcji stosować polecenie `\twocolumn`

lub `\twocolumn[Tekst]`. W tym drugim przypadku *Tekst* zostanie wypisany na początku strony przez całą jej szerokość. Oba polecenia umieszczone w tekście powodują zakończenie bieżącej jednokolumnowej strony (jak po `\clearpage`). Powrót do jednokolumnowego składu następuje po poleceniu `\onecolumn`. W miejscu *styl* może wystąpić tylko jedno z następujących słów:

<code>article</code>	oznacza format artykułu
<code>report</code>	— raportu,
<code>book</code>	— książki,
<code>letter</code>	— listu, oraz
<code>man</code>	— format podręcznika T _E X-a.

Użytkownik, szczególnie polski, przygotuje sobie zapewne swoje prywatne style opierając się na oryginalnych plikach (`article.sty`, `report.sty` ..., albo — znacznie wygodniejszych — `article.doc`, `report.doc` itd.) nadając im odmienne nazwy (np. `particle.sty`, `preport.sty` itd.) w celu uzyskania polskojęzycznych napisów generowanych automatycznie (takich jak np. „Spis treści”) oraz dla zadowolenia indywidualnego gustu w zakresie estetyki końcowego produktu. Nowo przygotowane style wczytuje się tak samo, jak te standardowe — poprzez nazwę umieszczaną w polu *styl*. Podobnie można postępować z plikami opcji (`art10.sty`, `art11.sty`, `art12.sty`, `rep10.sty` itd.). Zmodyfikowane i nowe pliki stylów i opcji zaleca się umieszczać w katalogach prywatnych, by nie „zaśmiecać” standardowego zestawu znajdującego się w domyślnym katalogu `\texinput` (T_EX szukając wskazanego pliku przegląda najpierw katalog bieżący, a dopiero potem ten domyślny).

Przed wpisaniem naszego zasadniczego tekstu możemy chcieć dodatkowo określić style i pewne parametry właściwe tylko do aktualnie przygotowywanej publikacji wczytując odpowiednie zbiory (poleceniem `\input nazwa.ext`, gdzie `.ext` jest rozszerzeniem nazwy pliku `nazwa` i można je pominąć, jeśli jest to `.tex`), albo wpisując te zbiory literalnie bezpośrednio po instrukcji `\documentstyle`. Np. polecenie

```
\nofiles
```

oznacza, że nie będą tworzone żadne dodatkowe zbiory poza `plik.dvi` i `plik.log` (ten pierwszy zawiera zakodowany skład całego tekstu niezależny od rodzaju urządzenia zewnętrznego; stąd rozszerzenie `.dvi` od ang. DeVice Independent, a w zbiorze `plik.log` znajdują się wszystkie komunikaty T_EX-a wysłane w czasie przetwarzania `plik.tex`). Bez polecenia `\nofiles` T_EX generuje dodatkowo zbiór `plik.aux` (zawierający informacje do poleceń `\ref` i `\pageref`, o których piszemy dalej) i ewentualnie inne w zależności od zakresu wykonywanych operacji.

Sposób formatowania żywej paginy (tj. nagłówek) i stopek na wszystkich stronach można określić poleceniem

```
\pagestyle{style},
```

w którym *style* to:

```
plain, empty, headings albo myheadings,
```

odpowiadające kolejno prostemu numerowi na środku dolnego marginesu strony tekstu (`plain`), pominięciu numeracji stron, standardowym nagłówkom i nagłówkom określonym

przez użytkownika. W tym ostatnim przypadku, tj. przy `myheadings`, używamy własnych opisów stosując:

```
\markboth{lewa główka}{prawa główka}, albo
\markright{prawa główka}.
```

(Dostępna jest także komenda dla wybranej pojedynczej strony:

```
\thispagestyle{style},
```

którą trzeba jednak umieścić w odpowiednim miejscu po `\begin{document}`).

Sposób wypisywania numerów stron ustalamy poleceniem

```
\pagenumbering{typ},
```

gdzie za *typ* możemy podstawić jedno z następujących słów:

```
arabic, roman, alph, Roman lub Alph,
```

które czynią numerację cyframi arabskimi, małymi rzymskimi, małymi literami antykwy (tj. minuskułami pisma rzymskiego), zwykłymi cyframi rzymskimi lub majuskułami antykwy. Np. strona siódma miałaby w kolejnych opcjach następujące postacie: 7, vii, g, VII lub G.

Dla usytuowania wydrukowanej strony względem brzegów kartki (i ekranu monitora) przydatne są polecenia typu: `\voffset=-1.1in` `\hoffset=-2.5cm`, omówione niżej.

Zasadniczy tekst naszej publikacji czy dokumentu wpisujemy po komendzie

```
\begin{document},
```

a cały plik kończy polecenie

```
\end{document}.
```

Możemy wpisać cokolwiek po tym ostatnim rozkazie, jednak \TeX w ogóle nie będzie tego analizował.

Stronę tytułową możemy (oczywiście nie jest to obowiązkowe) zredagować poleceniami:

- `\title{...}` — tytuł z wierszami oddzielonymi przez `\\`,
- `\author{...}` — autorzy oddzieleni przez `\and`,
- `\date{...}` — data uwidaczniana na dokumencie, oraz
- `\maketitle`, które korzystając z informacji z poprzednich trzech poleceń redaguje gotową stronę tytułową. Alternatywnie, stronę taką możemy zaprojektować całkiem według własnego uznania:

```
\begin{titlepage}
Material strony tytułowej ze składnikami rozdzielonymi znakami \\
\end{titlepage}
```

Podział dokumentu (tekstu) na **sekcje** odbywa się za pomocą poleceń, których argumentem jest tytuł sekcji, np. `\chapter{Tytuł}`. Oto pełny zestaw dostępnych sekcji:

```
\part      — część
\chapter   — rozdział
```

<code>\section</code>	— punkt
<code>\subsection</code>	— podpunkt
<code>\subsubsection</code>	— „podpodpunkt”
<code>\paragraph</code>	— paragraf
<code>\subparagraph</code>	— subparagraf

Sekcje mogą występować także w dodatku — po poleceniu `\appendix`. Każdą nazwę sekcji można opatrzyć znakiem `*` wskazującym, że nie ma być ona numerowana, np. `\chapter*{Tytuł}`. Głębokość numeracji sekcji ustala się przez nadanie parametrowi `\secnumdepth` odpowiedniej wartości (np. `\setcounter{secnumdepth}{2}` będzie znaczyło, że `\subsection` będzie jeszcze numerowane, ale `\subsubsection` już nie).

Spis treści (spis sekcji bez `*`) otrzymuje się po

```
\tableofcontents,
```

które powoduje, że tworzony jest zbiór z rozszerzeniem `.toc` (w naszym przypadku `plik.toc`), który w następnym „kompilowaniu” jest wykorzystany do zredagowania spisu w miejscu umieszczenia polecenia. Spis ten można uzupełniać o sekcje nie występujące *explicite* w tekście pisząc np.

```
\addcontentsline{toc}{section}{Tytuł sekcji},
```

gdzie za `section` można oczywiście podstawić też `part`, `chapter`, `subsection`, ... W powyższym poleceniu `Tytuł` można poprzedzić bezpośrednim podaniem numeracji: `\protect\numberline{numer}`. Ponadto, do spisu treści można także bezpośrednio dopisać dowolny tekst za pomocą polecenia

```
\addtocontents{toc}{Dodawany tekst}
```

Inne niż w tekście brzmienie tytułu w spisie zapewnia opcjonalny argument sekcji, np.

```
\section[Tytuł do spisu i żywej paginy]{Tytuł sekcji}.
```

Głębokość spisu reguluje parametr n , liczba, w poleceniu `\setcounter{tocdepth}{n}`.

Polecenia `\listoffigures` i `\listoftables` w podobny sposób generują spisy tabel i rysunków, przy czym w roli `toc` występują rozszerzenia i argumenty `lof` i `lot`, odpowiednio.

4. Ważne subtelności

Jeśli po słowach złożonych czcionkami pochyłymi następuje tekst złożony prostą czcionką, to przerwa międzywyrazowa jest krótsza niż zwykle — zależnie od ostatniej litery pochyłej. T_EX-owe polecenie `\/`, tzw. „korekta italikowa”, usuwa tę wadę odpowiednio do litery. Porównajmy:

<i>Kilof</i> los	<i>Graf</i> ;	<i>DOM</i> MODY
<i>Kilof</i> los	<i>Graf</i> ;	<i>DOM</i> MODY

W tym drugim przypadku napisaliśmy `\it Kilof\/ \rm los, \sl Graf\/\rm`; oraz `\it DOM\/ \rm MODY`. We wszystkich krojach każda litera posiada własną korektę italikową, chociaż zwykle w niepochyłych krojach jest ona zerowa. Są jednak wyjątki, np. `...f'`

trzeba zapisać jako `{\bf ...f\}`. Korekta italikowa jest zbędna, gdy po ostatniej literze pochyłej występuje kropka lub przecinek.

Inną specjalną poprawkę (zwiększenie odstępu) \TeX wprowadza automatycznie po kropkach kończących zdanie. Jeśli wewnątrz zdania wystąpi kropka z odstępem poprzedzona małą literą lub cyfrą (często jest to skrót typu „tzn.”) to, ażeby uniknąć wydłużania odstępu, należy po kropce postawić „wymuszony” odstęp: `_` albo `~` (tylda). Porównajmy skrót np. napisany tutaj zwyczajnie i np. zapisane jako `np._zapisane`. W sytuacji, gdy kropka kończąca zdanie nie jest poprzedzona małą literą ani cyfrą, \TeX nie zrobi zwykłej korekty. Ażeby ją wymusić, należy ową kropkę poprzedzić poleceniem `\@`. Jeśli nie zależy nam na tych niewielkich poprawkach, to możemy polecić `\frenchspacing` na początku dokumentu uzyskując jednakowe odstępy (przywrócenie poprawek następuje po `\nonfrenchspacing`).

Często spotykanym błędem jest pisanie krótkich kresek (-) tam, gdzie jesteśmy przyzwyczajeni widzieć nieco dłuższe (– lub —). Nawet wytrawni użytkownicy \TeX -a zdają się zapominać o tym, że te dłuższe kreski wpisuje się przez podwojenie lub potrojenie znaku łącznika: `--` lub `---`.

5. Jednostki \TeX -a

Niekiedy wypada powiedzieć \TeX -owi jak wiele ma przesunąć, wciąć lub opuścić określony element typograficzny. W tekście używamy określenia *dim* (od ang. dimension — wymiar), za które można wstawić pewne liczby ze znakiem, kropką lub przecinkiem dziesiętnym i jednostkami, przy tym podawanie jednostek jest obowiązkowe (można je wpisać dużymi lub małymi literami), zaś liczbę można opuścić (co jest równoważne 0) albo podać w systemie oktalnym (ósemkowym) poprzedzając znakiem `'`, bądź heksadecymalnym (szesnastkowym) poprzedzając znakiem `"`. Równoważne są więc formy: `15,0 pt` `'17pT` `"F pt` `+15.pt` `15Pt`. Pełny zestaw pierwotnych jednostek \TeX -a zawiera poniższa tabela.

Punkt (ang. point)	1 pt \cong 0,35146 mm
Pika (pica)	1 pc = 12 pt
Cal (inch)	1 in = 72,27 pt
Duży punkt (big point)	72 bp = 1 in
Centymetr	2,54 cm = 1 in
Milimetr	10 mm = 1 cm
Punkt drukarski (Didot point)	1157 dd = 1238 pt
Cycero (cicero)	1 cc = 12 dd
Skalowany punkt (scaled point)	65536 sp = 1 pt

6. Rozmiary i kroje czcionek (fonty)

\LaTeX posiada duży wybór krojów i rozmiarów czcionek łatwo osiągalnych przy redakcji tekstów. **Rozmiary** symboli wybiera się poleceniami:

`\tiny` — najmniejszy,
`\scriptsize` — wskaźnikowy,

`\footnotesize` — do przypisów,
`\small` — mały,
`\normalsize` — normalny,
`\large` — duży,
`\Large` — nieco większy,
`\LARGE` — **jeszcze większy** oraz
`\huge` — **największy**.

Przy niektórych opcjach możemy mieć dostęp do jeszcze jednego rozmiaru poprzez polecenie `\Huge`. Obszar w zakresie którego działają te polecenia (tak, jak wszystkie inne w \TeX-u) możemy ograniczyć nawiasami klamrowymi.

W trybie matematycznym (między znakami $\$ \dots \$$) mamy ponadto

`\mit` — *math italic*,
`\cal` — *CALLIGRAPHIC, TYLKO MAJUSKUŁY*,
`\displaystyle` — *displaystyle*,
`\textstyle` — *textstyle*,
`\scriptstyle` — *scriptstyle* oraz
`\scriptscriptstyle` — *scriptscriptstyle*.

Różnica między `\displaystyle` a `\textstyle` objawia się w sposobie składu i w wielkości czcionek użytych w „piętrowych” wyrażeniach matematycznych; pierwszy jest domyślny w konstrukcjach wyróżnionych z tekstu $\$ \$ \dots \$ \$$, a drugi — w tekstowych formułach postaci $\$ \dots \$$.

Pożądaną **krój** czcionki wybieramy z repertuaru:

`\rm` — ang. roman, czyli popularna antykwa lub krój romański zwany też rzymskim.
`\bf` — **BoldFace**, czyli druk tłusty (**pogrubięte pismo rzymskie**),
`\sc` — SMALL CAPS, CZYLI KAPITALIKI (ZMNIEJSZONE MAJUSKUŁY),
`\sf` — Sans-serif, bez zdobników (szeryfów) albo pismo blokowe,
`\it` — *italic*, czyli kursywa zwana też *italiką albo łacinką*,
`\sl` — *slanted*, czyli druk rzymski *pochylony*,
`\tt` — typewriter, czyli *naśladowujący maszynę*
`\uit` — upright italic, czyli *wyprostowana kursywa*

Warto zapamiętać, że każde wywołanie rozmiaru stowarzyszone jest z ustaleniem kroju na antykwę (jak po `\rm`). Rzeczywista wielkość czcionek zależy od opcji użytej w pierwszym poleceniu naszego pliku źródłowego (brak odpowiedniej opcji, 11pt lub 12pt). Relacje pokazuje tabelka obok, w której rzeczywiste wielkości podano w \LaTeX-owej konwencji małych liczb rzymskich z przyrostkiem „pt”. Np. `\xxvpt` oznacza wielkość 25 pt.

Rodzaje i rozmiary czcionek spoza podstawowego repertuaru \LaTeX-a wybieramy sekwencją typu:

```

\font\naszanazwa=cmfont scaled 1440,
\font\naszanazwa=cmfont \magstepm lub
\newfont{\naszanazwa}{cmfont \magstepm},
  
```

gdzie `\naszanazwa` oznacza przypisanie nowemu krojowi i rozmiarowi nazwy, `cmfont` jest nazwą kompletnego zestawu znaków wskazanego kroju w jednym rozmiarze, czyli tzw. *fontu* (np. `cmr10`). Fonty \TeX-a umieszczone są na ogół w katalogach `\pixel\dpiN` lub

Odpowiedniość rozmiarów i wielkości czcionek

Rozmiar \ opcja	(10pt)	11pt	12pt
\Huge	\xxvpt	\xxvpt	\xxvpt
\huge	\xxpt	\xxpt	\xxvpt
\LARGE	\xviipt	\xviipt	\xxpt
\Large	\xivpt	\xivpt	\xviipt
\large	\xipt	\xipt	\xivpt
\normalsize	\xpt	\xpt	\xipt
\small	\ixpt	\xpt	\xpt
\footnotesize	\viiipt	\ixpt	\xpt
\scriptsize	\vipt	\viiipt	\viiipt
\tiny	\vpt	\vipt	\vipt

... $\backslash N$ pt. W innych konfiguracjach katalogi te zamiast $\backslash \text{dpi}N$ miewają nazwy $\backslash \text{pxl}M$, gdzie $M = 5N$. Dobrze jest wiedzieć, że kolejne wartości N określa wzór $r \cdot 1,2^m$, gdzie $m = 0, 1/2, 1, 2, 3, 4$ lub 5 , a r jest rozdzielczością urządzenia drukującego w kropkach na cal (ang. *dots per inch* i stąd dpi). Najczęściej można spotkać trzy wartości dla rozdzielczości: 240 — drukarki 9-igłowe, 180 — drukarki 24-igłowe oraz 300 dla drukarek laserowych. Słowo kluczowe `scaled` jest nieobowiązkowe, a liczba 1440 jest 1000-krotną wartością powiększenia jakie chcemy przypisać nowemu fontowi względem podstawowego (tj. względem rozmiaru w katalogu `pxl1200`, albo `dpi240`, dla drukarki typu Epson, np. STAR NX-15). W wypadku liczby 1440 będzie to rozmiar o czynnik 1,44 większy od tego z `pxl1200` czyli zostanie załadowany wskazany font (a właściwie jedynie jego format, gdyż T_EX-a nie interesuje graficzna forma symboli, lecz tylko ich parametry) z katalogu `pxl1728` (albo `dpi346`).

Komplet znaków w foncie romańskim (cmr10)

	0	1	2	3	4	5	6	7	8	9
x	Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ
1x	Ω	ff	fi	fl	ffi	ffl	ı	ı	ˆ	ˆ
2x	˘	˘	-	˚	˘	ß	æ	œ	ø	Æ
3x	Œ	Ø	-	!	”	#	\$	%	&	,
4x	()	*	+	,	-	.	/	0	1
5x	2	3	4	5	6	7	8	9	:	;
6x	ı	=	ı	?	@	A	B	C	D	E
7x	F	G	H	I	J	K	L	M	N	O
8x	P	Q	R	S	T	U	V	W	X	Y
9x	Z	[“]	^	·	‘	a	b	c
10x	d	e	f	g	h	i	j	k	l	m
11x	n	o	p	q	r	s	t	u	v	w
12x	x	y	z	-	—	”	˘	˘		

Dla powiększenia połówkowego zamiast `\magstep` z cyfrą piszemy `\magstephalf`. Fragment tekstu wypisywany określonym tak fontem wystarczy poprzedzić teraz łań-

Komplet znaków w foncie cmsy10

	0	1	2	3	4	5	6	7	8	9
x	–	·	×	*	÷	◇	±	≡	⊕	⊖
1x	⊗	⊙	⊛	⊜	⊝	●	×	≡	⊃	⊄
2x	∠	∩	∪	∩	∪	≈	∩	∪	⊂	⊃
3x	∩	∪	↑	↓	↑	↓	↔	↗	↘	↔
4x	↔	↔	↕	↕	↕	↘	↙	∞	/	∞
5x	∈	∉	△	▽	/		∇	∃	¬	∅
6x	ℜ	ℑ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ	ℓ
7x	ℱ	ℱ	ℱ	ℱ	ℱ	ℱ	ℱ	ℱ	ℱ	ℱ
8x	ℙ	ℚ	ℛ	ℜ	ℜ	ℜ	ℜ	ℜ	ℜ	ℜ
9x	ℤ	ℤ	ℤ	ℤ	ℤ	ℤ	ℤ	ℤ	ℤ	ℤ
10x	⌈	⌋	{	}	<	>			↕	↕
11x	\	˘	√	∏	∇	f	∩	∩	∩	∩
12x	§	†	‡	¶	♣	◇	♥	♠		

cuchem `\naszanazwa` po czym przywrócić poprzedni font np. `\rm` lub ująć ów fragment wraz z łańcuchem w nawiasy klamrowe: `{, }`. Tutaj mamy italikę 7-punktową (`cmti7`) uzyskaną sekwencją: `{\font\tiseven=cmti7 \tiseven` Tutaj ...`}`.

Odpowiedniość powiększeń
i nazw katalogów fontów

<code>\magstep...</code>	0	half	1	2	3	4	5
Powiększenie	1000	1095	1200	1440	1728	2074	2488
ELQ							
dpi...	180	197	216	259	311	373	448
pxl....	900	986	1080	1296	1555	1866	2239
Epson							
dpi...	240	263	288	346	415	498	597
pxl....	1200	1315	1440	1728	2074	2488	2986
laser							
dpi...	300	329	360	432	518	622	746
pxl....	1500	1643	1800	2160	2592	3110	3732

Trzeba wspomnieć, że fonty doładowywane w ten sposób nie mogą być używane w trybie matematycznym.

Każdy ze znaków zestawu można umieścić w tekście pisząc `\charNr` albo `\symbol{Nr}`, gdzie `Nr` jest jego numerem dziesiętnym. Np. `\char71` powoduje wydrukowanie `G` z aktualnie używanego kroju (tu rzymski 12pt, tj. `cmr12`). Jeśli posiadamy tabele fontów ponumerowanych w systemie ósemkowym albo szesnastkowym, to wygodniej będzie nam pisać `Nr` w tym właśnie systemie poprzedzając wpisywaną liczbę znakiem `'` (apostrof, dla liczb oktalnych) lub cudzysłowem `"` (dla heksadecymalnych).

Więcej informacji o czcionkach przedstawiamy w punkcie 22. (od s. 47).

7. Redakcja tekstu — strona, paragraf

Położenie lewego górnego rogu złożonego tekstu na stronie w L^AT_EX-u definiują komendy:

`\hoffset=dim` i `\leftmargin dim` — poziomo,
`\voffset=dim` i `\topmargin dim` — pionowo.

(Powtarzające się często w komendach `h` i `v` są skrótami od ang. *horizontal* i *vertical*.)
 Rozmiar kolumny tekstu zmieniamy rozkazami:

`\hsize=dim` lub `\textwidth dim`, — szerokość,
`\vsize=dim` lub `\textheight dim` — wysokość.

△ Skład tekstu T_EX wykonuje **paragrafami**. Paragraf zwykle zaczyna się od wcięcia. Jeśli nie chcemy wcięcia, to piszemy w danym miejscu rozkaz

`\noindent;` `\indent`

zaś wymusza wcięcie na początku wiersza nie zaczynającego paragrafu. Rozmiar wcięcia możemy zmienić komendą

`\parindent=dim`,

gdzie *dim* jest wymiarem w jednostkach długości, np. `-3cm`, `+1,3in`, `27 pt` (`1pt` \cong `0,35146mm`) itp. W obszar wcięcia można wstawić jakiś materiał poleceniem

`\textindent{material}`.

Np. ten paragraf rozpoczęliśmy od następującego łańcucha `\textindent{\$triangle$}`
`Skład ...`

Polecenie `\hang` powoduje, że wcięcie dotyczy całego paragrafu, a nie tylko jego pierwszego wiersza. Ogólniej jednak mamy dwa polecenia regulujące głębokość wcięcia w obu kierunkach i jego położenie w paragrafie:

`\hangindent=dim` oraz `\hangafter n`,

gdzie *dim* ma wymiar długości i jeśli jest dodatni, to wcięcie będzie z lewej strony paragrafu, w przeciwnym zaś razie — z prawej. Liczba *n* określa od którego wiersza ma się rozpocząć wcinanie paragrafu. Jeśli *n* > 0, wtedy wcięte będą wszystkie wiersze poczynając od (*n* + 1)-szego. Jeśli *n* < 0, to wciętych zostanie $|n|$ pierwszych wierszy. Następny paragraf poprzedza `\hangindent-1cm\hangafter 6`.

Paragraf kończy rozkaz `\par` lub pusta linia (ewentualne dalsze puste linie są ignorowane, podobnie jak kolejne odstępy po pierwszym). Linia skończy się przed prawym marginesem jeśli napiszemy `\`, `\break` albo `\newline` — tak, jak w tym miejscu. (Ten sam `\break` po `\par` łamie stronę.) Po `\` możemy podać `*`, co oznacza tyle co „nie łam strony po tym wierszu”, lub/ oraz [*dim*] zwiększające odstęp do następnego wiersza o *dim* (np. `*[2mm]`). Podobny efekt daje polecenie `\adjust{\kerndim}` — umieszczone wewnątrz wiersza odsuwa następny o *dim*. Jeśli chcemy mieć pewność, że dana strona zakończy się bieżącym wierszem, to umieszczamy w nim `\adjust{\eject}`. Dostępne są też polecenia `\linebreak[num]` i `\nolinebreak[num]`, gdzie *num* ozna-

cza priorytet (wagę) i może mieć wartość od 0 (najniższy) do 4.

Odstęp między paragrafami możemy ustawić rozkazem `\parskip`, np.

```
\parskip1.0pt plus 2pt
```

oznacza odstęp od 1 do 3 punktów (dobierane przez T_EX-a). Jednorazowo możemy zmienić odstęp pisząc po `\par \vspace{dim}`. Wersja `\vspace*{...}` działa także w miejscu złamania strony. Wygodne są też standardowe odstępy:

```
\smallskip = \medskip = \bigskip =
```

Ten paragraf kończy `\vspace {5mm}`.

Dodatkowy lewy lub prawy **margins** wprowadzamy rozkazami:

```
\leftskipdim lub
\rightskipdim,
```

odpowiednio. Zakres działania tych komend możemy ograniczyć obejmując wybrane paragrafy (wraz z poprzedzającymi komendami) w nawiasy klamrowe: `{ i }`, bądź wołając ponownie te rozkazy z *dim* równym 0 mm. Ten paragraf został poprzedzony sekwencją `\leftskip 15mm \rightskip1.5cm`.

Podobny efekt uzyskuje się poleceniem `\narrower`, które zawęży kolumnę z obu stron o wartość `\parindent`. Kolejne użycia `\narrower` akumulują się.

Brzegi paragrafu możemy także formatować w sposób dowolny używając polecenia

```
\parshape= n i1 l1 i2 l2 ... in ln,
```

gdzie i_k oznacza wielkość wcięcia k -tego wiersza o długości l_k . Taki format kasuje `\parshape=0`. Np., ten paragraf rozpoczęliśmy od: `\parshape = 3 1.2cm 9cm .9cm 9.3cm .6cm 9.6cm \noindent`.

Jeśli paragraf zakończy się prawie pustą linią (2–3 samotne znaki we wierszu nie sprawiają zbyt dobrego wrażenia), to by zmienić sposób złożenia tego paragrafu możemy podać polecenie `\looseness=n`, gdzie n może być większe lub mniejsze od zera. Po takim poleceniu T_EX spróbuje rozciągnąć ($n > 0$) lub scieśnić ($n < 0$) paragraf o $|n|$ wierszy.

Odstępy wierszy określają style i opcje. W szczególności, każde wywołanie innego rozmiaru czcionki ustala tą wartość w jednostce `\baselineskip` — jest to minimalny odstęp do następnego wiersza. Nie zaleca się zmieniać tej wielkości, chociaż można to uczynić. Możemy natomiast swobodnie przeskalowywać wszystkie odstępy o czynnik α pisząc w preambule dokumentu `\renewcommand {\baselinestretch}{\alpha}`. Odstępy wyrazów we wierszu można regulować rozkazem `\hspace{dim}` (np. ten odstęp powstał z `\hspace{1cm}`). Jeśli taki rozkaz przypadnie na zmianie wiersza, wtedy nie będzie efektywny. Możemy jednak wymusić uwzględnienie go poprzez umieszczenie rozkazu w tzw. *hbox*, np. `\hbox{\hspace{5em}}` (1em to szerokość litery „M”). Wersja `\hspace*{...}` działa także na początku wiersza. Dowolny tekst w *hbox* pozostaje w

jednym wierszu — nawet gdy jest dłuższy niż `\hsize`. Poziome odstępy to także:

„Miękkie”		„Twarde”
<code>\,</code> (1/6 em)		<code>\thinspace</code>
<code>_</code> (1/3 em)		<code>~</code> (tylda)
<code>\enskip</code> (0,5em)		<code>\enspace</code>
<code>\quad</code> (1em)		
<code>\qqquad</code>		<code>\negthinspace</code>
<code>\hskip</code> <i>dim</i>		<code>\kern</code> <i>dim</i>

przy czym na „miękkich” odstępach \TeX może łamać wiersz, zaś na „twardych” — nie.

Odstęp `\negthinspace` jest ujemny (i wynosi $-1/6$ em), `\qqquad` oznacza podwojenie `\quad`, a `\hskip` i `\kern` mają wartość *dim*. Możemy też zdefiniować własną jednostkę długości, np. o nazwie `\dlu`, za pomocą poleceń:

`\setlength{\dlu}{dim}` lub `\newlength{\dlu}` i `\settowidth{\dlu}{Co/s}`

(w drugim przypadku jednostce `\dlu` zostanie przypisana wartość równa szerokości złożonego tekstu „Co”).

Normalnie tekst jest wyrównywany przy obu marginesach. Rozkaz `\raggedleft` daje efekt taki, jak w poprzednim paragrafie (nieregularny prawostronny margines dostaniemy po `\raggedright`). Przed bieżącym paragrafem napisaliśmy `\flushleft`; jest także polecenie `\flushright`, działające w odwrotną stronę. W przypadku wymienionych poleceń ostatni paragraf, którego ma dotyczyć rozkaz należy zakończyć poleceniem `\par` lub pustą linią, a dopiero po nim postawić zamykający całość nawias klamrowy. ←

Analogiczne do poziomych, polecenia `\raggedbottom` (przeciwnie: `\flushbottom`), `\newpage`, `\pagebreak [num]`, `\nopagebreak[num]` dotyczą miejsca złamania strony (między paragrafami). \TeX na ogół dobrze łamie strony, lecz bywa że przesuwa niepotrzebnie ostatni wiersz akapitu kończącego stronę na następną stronę. W takich przypadkach, gdy chcemy by paragraf pozostał cały na jednej stronie, piszemy

`\samepage.`

Polecenie to zabrania złamania strony także między akapitem, a materiałem eksponowanym (np. wzorem matematycznym). W ogólności, przy niewłaściwych złamaniach strony piszemy wspomniane polecenie w odpowiednim paragrafie, używamy `\nopagebreak` po pustych wierszach, zaś `\pagebreak` — w dopuszczalnych miejscach.

Oprócz wymienionych już parametrów stylów (takich jak `\parindent` czy `\baselineskip`) istnieje wiele innych, spośród których niektóre można dość łatwo zmieniać w dowolnym miejscu. Oto kilka z nich wraz z wartościami im przypisanymi w `art10.sty` (tj. domyślnym zbiorze opcji stylu `article.sty`):

`\oddsidemargin 44pt` (63pt, gdy bez opcji `twoside`) — lewy margines stron nieparzystych

`\evensidemargin 82pt` (63pt) — j.w. dla stron parzystych (oba liczą się od miejsca odległego o jeden cal od brzegu kartki)

`\marginparwidth 107pt (90pt)` — szerokość obszaru do notek na marginesie. Notki można wpisać poleceniem `\marginpar[Na lewej str.]{Na prawej str.}`, gdzie opcjonalny argument jest używany, gdy polecenie to wypadnie na lewej stronie (tutaj i na poprzedniej stronie napisaliśmy `\marginpar [\hfill \Rightarrow] {\Leftarrow}`)
`\marginparsep 11pt` — odległość wyżej opisanego marginesu od tekstu
`\topmargin 27pt` — odległość od górnego brzegu kartki do żywej paginy minus 1 cal
`\headheight 12pt` — wysokość żywej paginy (nagłówek)
`\headsep 25pt` — odległość żywej paginy od tekstu
`\footskip 30pt` — odległość stopki strony od tekstu
`\footnotesep 6.65pt` — odległość przypisów od tekstu
`\columnsep 10pt` — odległość kolumn na stronach dwukolumnowych
`\columnseprule 0pt` — grubość linii pionowej rozdzielającej kolumny przy opcji `twocolumn` (0pt oznacza, że linii nie ma!)

8. Bloki i ramki (*boxes*)

Praca T_EX-a, podobnie jak zecera i metrapaży, polega na składaniu „klocków” — cegiełek i bloków reprezentujących wymiarowo symbole i inne obiekty typograficzne. Każdy symbol (w fontach) jest przez T_EX-a traktowany jako blok o określonej tylko dlań wysokości (`\ht`), szerokości (`\wd`) i głębokości (`\dp`) (ten ostatni parametr mówi jak głęboko poniżej podstawy składanego wiersza wystaje dany znak). Np. litery „M j” traktowane są dosłownie jak „□ j”. Nasz

T_EX składa takie kostki w wiersze tak, jak widzimy tutaj:



(rysunek ten należy porównać z wierszem powyżej), wstawiając pomiędzy wyrazy, w miejscach odstępów, sprężyste twory („lepiszcze”) zwane *glues*, które pozwalają mu dopasowywać długość wierszy do zadanej wartości. Takim tworem jest np. `\hskip 1cm plus 2cm minus 5mm`, a ogólniej:

```

\hskip dim plus dim minus dim
\vskip dim plus dim minus dim

```

Kiedy T_EX składa paragrafy z wierszy mówimy, że jest w trybie *horyzontalnym*. Gotowe paragrafy stanowią znów bloki, które T_EX układa w pionie (stąd tryb *wertykalny*) wstawiając pomiędzy bloki podobne do poziomych twory elastyczne w kierunku góra–dół (np. `\vskip...`). (W L^AT_EX-u rozróżnia się tryb LR, składanie wierszy z lewa na prawo, i akapitowy — składanie paragrafów).

Wspomniany wcześniej *hbox*, to kostka, w którą można wstawić cokolwiek, a T_EX uszereguje to starając się rozłożyć elementy składowe możliwie równomiernie na całej jego długości. Ma ona też formy:

```

\hbox to dim {tekst} oraz
\hbox spread dim {tekst}.

```

Np. `\hbox to \hsize {tekst}` rozkłada `tekst` na całej długości wiersza. Druga forma

rozluźnia tekst o dim względem jego naturalnej długości (tzn. wynikającej z prostej sumy szerokości znaków i nieściśniętych ani nie rozciągniętych odstępów międzywyrazowych). Pionowym odpowiednikiem $hbox$ -u jest $vbox$ ($\vbox{\dots}$), którego argumentem może być wiele wierszy, a nawet kilka paragrafów.

Pożytecznym będzie uświadomić sobie, że elementy ($boxy$) składające się na $hbox$ T_EX ustawia szeregowo, jeden obok drugiego w poziomie, natomiast w $vboxie$ wyrównywanie odbywa się względem pionu — jeden pod drugim. Wszystkie $boxy$ są umiejscawiane w rejestrach T_EX-a zwanych $\box N$, gdzie N przyjmuje wartość z zakresu 0 – 255, przy czym do bezpośredniego wykorzystania przez użytkowników pozostawiono wolne numery od 0 do 9, a pozostałe rejestry wykorzystuje system. Wymiary $boxów$ są dostępne poprzez wspomniane wyżej parametry i numer (np. $\wd 5$) i możemy je zmieniać i wykorzystywać jako dim . Np. po $\setbox 2=\hbox{()}\the\ht 2--\the\dp 2\copy 2$ otrzymujemy: 9.0pt–3.0pt(). Polecenie $\box 2$ powoduje umieszczenie zawartości tego $boxu$ w tekście z jednoczesnym opróżnieniem go: $=\box 2=\box 2=$ daje $=()=$.

Użyteczną odmianą $boxów$ jest

$\fbox{\dots}$,

gdzie \dots oznaczają zawartość $frameboxu$ i może to być $\vbox{\dots}$, jeśli zawartość \dots jest dłuższa niż \hsize (tj. nie mieści się w jednym wierszu). Np. Ramka ... dostajemy

pisząc $\fbox{Ramka \dots}$, zaś 1 2 3 4
5 6 7 8 9 10 — $\fbox{\hsize 2cm \vbox{1 2 3 4 5 6 7 8 9 10}}$. Grubość linii obramowania i jej odległość od tekstu regulujemy przypisując nowe wartości parametrom

\fboxrule (domyślnie 0,4 pt) i
 \fboxsep (domyślnie 3 pt);

np. Ramka ... dostajemy po sekwencji $\fboxrule=.5mm \fboxsep=3mm \fbox{Ramka \dots}$. Polecenie

$\dashbox{dług}(x,y)\{zawarto/s/c\}$

daje obramowanie linią przerywaną.

Przykładowo ramkę: centrowana zawartość ramki, dostaliśmy pisząc $\dashbox{5}(220,16)\{centrowana\ zawarto/s/c\ ramki\}$, gdzie trzy argumenty to: długość kresiek (tu 5pt), szerokość ramki (220pt) i wysokość ramki (16pt). Jednostki argumentów (domyślnie punkty) można zmienić na inne, np. na milimetry pisząc $\unitlength=1mm$. Ogólniej mamy $\dashbox\{(\,)[poz]\{\dots\}$, gdzie poz oznacza umiejscowienie zawartości ramki: t = top (góra), b = bottom (dół), l = left (lewo), r = right (prawo), c = center (środek; parametr domyślny), tl = top left (góra z lewej) i tej podobne kombinacje. Ramka owalna powstaje po $(\put(-85,2.5)\oval(180,15))$, gdzie argumentami są współrzędne środka owalu ($-85,2.5$) (licząc względem miejsca rozkazu \put — tutaj bezpośrednio po $(180,15)$) oraz jego szerokość i wysokość (w „ \unitlength -ach”).

Dla wielokrotnych powtórzeń pewnych sekwencji znaków bądź tworów typograficznych można zdefiniować nazwę \nazwa :

`\newsavebox {\nazwa}`, przypisać jej zawartość:
`\savebox{\nazwa}[dim]{tre/s/c}`

i wywoływać kiedy zachodzi potrzeba poprzez nazwę

`\usebox{\nazwa}`.

Polecenia te wykorzystują rejestry systemowe (o numerach większych od 9). Forma `\sbox N{...}` jest wygodnym skrótem od `\savebox N{...}`, gdzie `N` może mieć wartość między 0 i 255.

Dostępne są też polecenia:

`\makebox[szerokosc][l]{...}`, gdzie `[l]` oznacza dosunięcie do lewego skraju *boxu*, a `[r]` — do prawego

`\framebox ...` — wersja `\makebox` z ramką

`\fbox{...}` efektywnie jest skrótem dla `\framebox{...}` (tj. bez opcjonalnych argumentów)

`\mbox{...}` — efektywnie skrót dla `\makebox{...}`

`\raisebox{dim}[wysokosc][glebokosc]{...}`, gdzie *dim* może być dodatnie (przesunięcie w górę) lub ujemne (w dół)

`\showboxN` podaje zawartość wskazanego *boxu* (tj. `N`-tego) wpisując informację do pliku `.log`

`\parbox[t]{szerokosc}{...}` podstawia *szerokosc* do `\hsize` i buduje paragrafy z `...`; przy opcjonalnym argumente `[t]` powstały blok jest horyzontalnie wyrównywany względem górnego wiersza, przy `[b]` — względem dołu, zaś bez tego argumentu — względem środka

Działanie *parboxu* zilustrują następujące trzy bloki o szerokości 2 cm każdy, wpisane

w ten paragraf. Blok pierwszy

utworzony został <i>parboxem</i> z opcją <code>[t]</code>
--

, drugi

blok ma opcję <code>[b]</code>

, ostatni

zaś nie ma opcjo- nalnego ar- gumentu
--

(*parboxy* nie obramowują tekstu, ale tutaj dla wyrazistości ujęliśmy je same w *fboxy*). Analogicznie zachowuje się otoczenie `minipage` (s. 22).

9. Linie

W tekście możemy używać różnego rodzaju **linii**. Linie `-`, `—`, `—`, `—`, `—` uzyskuje się po `-`, `$- $\$$` , `--`, `---`, `\rule[-1.5mm]{20mm}{1mm}`, gdzie `\rule` ma trzy argumenty: wysokość dolnego skraju linii, długość i grubość. Pierwszą z wymienionych, łącznik zwany też dywizem albo tiretem, używa się zwykle do łączenia wyrazów dwuczłonowych (np. w XX-leciu) i przy przenoszeniu sylab wyrazów dzielonych na końcu wiersza. Drugi, `-`, jest matematycznym znakiem odejmowania (minus). Znak kodowany jako `--` występuje w zapisach przedziałów liczb (np. 2–3, albo 2 – 3). Wreszcie `—` (kodowane `---`) jest zwykłym myślnikiem (pauzą).

Linie przez całą szerokość strony narysujemy poleceniem `\hrule`:
 Ogólniej mamy:

```
\hrule widthdim heigt dim depthdim
\vrule widthdim heigt dim depthdim
```

Rozkaz `\vrule` kreśli linię pionową o wysokości danego *boxu*: |. W wierszu tekstu będzie to linia rozciągająca się od miejsca, dokąd sięga najwyższy znak do poziomu najniższego punktu we wierszu. Dynamiczną linię (wypełniającą dany *box* do końca) dostaniemy po `\hrulefill` _____.

10. Otoczenia (*environments*)

Otoczenia, zwane też „środowiskami”, przeznaczone są do wyróżnionego formatowania ograniczonego obszaru tekstu. W prostszym przypadku jest to efektywnie obszar otoczony nawiasami klamrowymi, na początku którego jest jedno polecenie (np. `\flushleft`). W ogólności są to konstrukcje typu:

```
\begin{otoczenie} ... \end{otoczenie},
```

a w niektórych przypadkach posiadające również argumenty. *Otoczeniem* może być:

- `quote` — tekst obustronnie wcięty
- `quotation` — tekst wcięty z wcięciami paragrafów
- `em` — tak, jak po poleceniu `\em`, tekst zostaje wyróżniony kursywą (`\it`) w druku pismem prostym, a antykwą (`\rm`) — w pochyłym
- `verse` — wiersz ze znakami `\\` kończącymi linie i zwrotkami oddzielonymi przez puste wiersze
- `center` — tekst centrowany z elementami oddzielonymi znakami `\\` (ogólniej: `*[dim]`)
- `flushright/flushleft` — tekst dosunięty do prawej/lewej strony
- `verbatim[*]` — tekst literalny czcionką maszynową (`\tt`). Wersja z gwiazdką, tj. `verbatim*`, używa znaków `\` zamiast odstępów. Dostępne jest też odpowiednie polecenie `\verb[*]|...|`, gdzie `|` może być dowolnym znakiem nie występującym wśród ...
- `abstract` — tekst streszczenia
- `theindex` — materiał na dwukolumnowy skorowidz uzyskiwany po `\makeindex`. Każdą pozycję rozpoczyna polecenie `\item`, a jeśli owa pozycja ma „podpozycje”, to poprzedza się je poleceniami `\subitem` i (jeszcze głębiej) `\subsubitem`
- `sloppypar` — luźne paragrafy (wyrazy są rzadziej dzielone na końcach wierszy kosztem tolerancji większych odstępów międzywyrazowych). Samo polecenie `\sloppy` działa tak samo, lecz dotyczy całego tekstu — aż do napotkania kasującego polecenia `\fussy`
- `math` — wzór matematyczny w tekście, jak po `$... $` lub `\(... \)`
- `equation` — wzór automatycznie numerowany z prawej strony (z lewej przy opcji `leqno`, dosunięty do lewej — przy opcji `fleqn`)
- `displaymath` — wzór bez numeracji, jak po `$$... $$` lub `\[... \]`

- `eqnarray` — trójkolumnowy układ automatycznie numerowanych równań (środkowa kolumna jest centrowana, otoczona znakami `&` i w stylu `\textstyle`, skrajne dosuwane do środkowej i w `\displaystyle`). Po wpisaniu `\nonumber` przy wybranym równaniu nie pojawi się tam numeracja. Poszczególne wiersze (równania) oddziela się znakami `\\` ewentualnie `*[dim]`, gdzie `*` (można pominąć) zabrania złamania tutaj strony, a `dim` jest dodatkowym odstępem wierszy
- `eqnarray*` — j.w., ale bez numeracji równań
- `itemize` — tekst zawierający segmenty oddzielone rozkazem `\item`, a po złożeniu tekstu — znakiem `•`
- `enumerate` — jak wyżej, ale z numeracją 1., 2., ... zamiast `•`
- `description` — jak wyżej, ale oznaczanie segmentów przez `\item[oznaczenie]`
- `minipage` — jest to otoczenie z argumentami. Pełna forma ma postać:

```

\begin{minipage}[t]{szer}
zawartość otoczenia; może być kilka akapitów
\end{minipage}

```

Tekst otoczenia jest formowany w paragrafy o szerokości `szer` i ustawiany horyzontalnie w zależności od opcjonalnego argumentu. Jeśli argument ma postać `[t]`, to złożony blok jest wyrównywany względem górnego wiersza, przy `[b]` — względem dołu, zaś bez tego argumentu — względem środka bloku. Ilustracja tego wyrównywania znajduje się przy opisie polecenia `\parbox` (s. 20), które zachowuje się bardzo podobnie. Środowisko to ma własny licznik przypisów. Jeśli na początku otoczenia umieszczamy materiał eksponowany (np. `$$... $$`), to powinniśmy poprzedzić to poleceniem `\vspace{-\abovedisplayskip}`, w celu usunięcia nadmiernego odstępu od tekstu.

Kilka dalszych otoczeń (`tabbing`, `tabular`, `table`, `picture` i `figure`) omawiamy w następnym punkcie.

11. Tabele i rysunki

Przy przygotowywaniu tabel mamy do wyboru kilka możliwości należących do kategorii otoczeń z argumentami. Omówimy je po kolei włącznie z $\text{T}_{\text{E}}\text{X}$ -owym `\halign`.

- `tabbing` — robi tabelkę z tekstu, w którym
 - `\=` ustawia tabulator,
 - `\>` oznacza skok w przód do takiego tabulatora,
 - `\+` — przesunięcie lewego marginesu o jeden tabulator w przód,
 - `\<` na początku wiersza kasuje w nim efekt `\+`,
 - `\-` kasuje trwale efekt jednego `\+`,
 - `\'` i `\'` dosuwa w prawo i lewo w kolumnie,
 - `\\` zmienia wiersz (można to rozszerzyć o `*[dim]`), zaś
 - `\kill` umieszczone zamiast `\\` powoduje opuszczenie tekstu wiersza, ale zachowanie tabulatorów.
- Są tu też polecenia `\pushtab` (zapamiętanie aktualnych pozycji tabulacji) i `\poptab` (odtworzenie zapamiętanych tabulacji).

Zwykle tekstowe akcenty \backslash' , \backslash' , $\backslash=$ w tym otoczeniu uzyskuje się pisząc $\backslash a'$, $\backslash a'$, $\backslash a=$. Oto prosty przykład (za Lamportem 1985):

If it's raining	$\backslash begin{\tabbing}$
then put on boots,	If $\backslash =it's$ raining $\backslash\backslash \backslash >then$
take hat;	$\backslash =put$ on boots, $\backslash\backslash$
else smile.	$\backslash >\backslash >take$ hat; $\backslash\backslash$
Leave house	$\backslash >else\backslash >smile.\backslash\backslash$ Leave house
	$\backslash end{\tabbing}$

Podobnemu celowi służy także pierwotna instrukcja T_EX-a $\backslash align$, ale jest szybsza od pozostałych. Jej składnia wygląda tak:

```
 $\backslash align$  [to dim] {col&col&...[&&col] $\backslash cr$ 
...&...&... $\backslash cr$ 
:
...&...&... $\backslash cr$ },
```

gdzie *col* określa format odpowiedniej kolumny (może być dla każdej innej) i musi zawierać znak #, w którego miejsce będzie podstawiona zawartość ... kolejnych wierszy. Poza tym znakiem w *col* mogą występować dowolne symbole i polecenia typu $\backslash hfil$ (użyteczne w celu dosunięcia do odpowiedniej strony kolumny; format $\backslash hfil#\backslash hfil$ centruje kolumnę). Jeśli przed *col* wystąpią znaki &&, to oznacza że format odtąd do $\backslash cr$ powtarza się cyklicznie w nieskończoność (jeśli && wypadnie przed pierwszą kolumną, to pisze się tylko &). Po $\backslash cr$ można napisać $\backslash noalign\{...\}$, gdzie ... stanowi materiał pionowy (np. $\backslash vskip 2mm$, $\backslash vspace \{2mm\}$, $\backslash medskip$). Także po $\backslash cr$ można podać $\backslash hline$ uzyskując linię poziomą. Przykładowo, tabelkę:

n	=	0	1	2	3	4	...
$\mathcal{F}(n)$	=	1	2	4	7	...	

```
uzyskamy pisząc:  $\$ \$ \backslash vbox\{$ 
 $\backslash align\{\$ \backslash hfil# =\$ \& \backslash \backslash hfil#\backslash hfil\backslash cr\backslash hline$ 
 $\backslash noalign\{\backslash smallskip\}$ 
 $n\backslash phantom\}&0&1&2&3&4&\dots\backslash cr$ 
 $\backslash noalign\{\backslash vskip.5mm\}$ 
 $\{\backslash cal F\}(n)\&1&2&4&7&\dots\backslash cr$ 
 $\backslash noalign\{\backslash vspace\{5pt\}\}\backslash hline\}$   $\$ \$$ 
```

Całość otoczyliśmy znakami $\$ \$$ w celu scentrowania tabelki na środku kolumny. Użyty tutaj $\backslash phantom\{Tekst\}$ zostawia przestrzeń, tj. odstęp, odpowiadającą tekstowi „Tekst”. Odstępy kolumn są automatycznie dobierane. Gdy chcemy je zmienić podajemy np. $\backslash tabskip=1em$ plus $2em$ minus $.5em$, albo „twardo” $\backslash tabskip=3pt$.

Kolejne konstrukcje mają formę otoczeń.

- $\backslash begin{array}\left[\begin{matrix} t \\ b \end{matrix} \right] \{format\}$
wiersze tabelki oddzielone znakami $\backslash\backslash$
 $\backslash end{array}$,

gdzie $[\begin{smallmatrix} t \\ b \end{smallmatrix}]$ reguluje sposób ustawienia kilku `array` obok siebie lub obok linii tekstu (dla otoczenia `tabular`). Można go całkowicie pominąć (tabelki ustawione środkami swych wysokości na jednej linii), albo napisać w to miejsce `[t]` (ustawianie według położenia górnego wiersza), albo `[b]` (według dolnego wiersza). *Format* jest ciągiem znaków (bez odstępów) składających się z `l`, `r`, `c` odpowiadających kolejno wyrównywaniu do lewego skraju kolumny, prawego skraju lub centrowaniu. Kolumny w wierszach oddziela się znakiem `&`. Otoczenie `array` może występować jedynie w trybie matematycznym. Kilka `array` może być zagnieżdżone jedno wewnątrz drugiego (przykład podajemy na str. 32).

```
■ \begin{tabular}[\begin{smallmatrix} t \\ b \end{smallmatrix}]{format}
  wiersz1 \\ (albo \\*[dim])
  wiersz2 \\ (albo \\*[dim])
  :
  wierszN \\
\end{tabular},
```

gdzie $[\begin{smallmatrix} t \\ b \end{smallmatrix}]$ — jak w `array`. Każdy *wiersz* składa się z elementów kolumn oddzielonych znakiem `&`. Sposób redagowania kolumn określa *format*, składający się z oznaczeń: `l` (dosunięcie do lewej strony kolumny), `c` (centrowanie), `r` (do prawej strony), `|` (pionowa kreska), `@{text}` (gdzie `text` może być np.: `\hspace{dim}`, `\extracolsep{dim}`, `\hfill`; forma `@{}` eliminuje odstęp międzykolumnowe, a zatem także te na skrajach tabeli bez obramowania, tj. bez znaku `|`). `n`-krotne powtórzenia formatu kolumny koduje się w postaci `*{n} {format}`. Linie poziome przez kilka kolumn rysuje się rozkazem `\cline{n1-n2}`, gdzie `n1` i `n2` są numerami skrajnych kolumn. Na całą szerokość tabeli wystarczy polecenie `\hline`; podwójne linie powstają po `\hline \hline`, a pionowe na całą wysokość wiersza — `\vline` (te ostatnie można „pchnąć” na skraj kolumny pisząc z odpowiedniej strony `\hfill`). W celu napisania tekstu przez `k` kolumn używamy polecenia:

```
\multicolumn{k}{format}{wstawiany material}.
```

Oto przykład (za Lamportem 1985):

<i>Type</i>	<i>Style</i>	
smart	red	short
rather silly	puce	tall

```
\begin{tabular}{|l|l|r|}
\hline\hline \em Type &
\multicolumn{2}{c|}{\em
Style}\ \ \hline smart
&red&short\ \ \cline{2-3}
rather silly&puce&tall\ \
\hline\hline \end{tabular}
```

Otoczenie `tabular` jest przez `TEX`-a traktowane jak pojedynczy wielki symbol (tak, że można je wstawić nawet między litery wyrazu!).

Wartości parametrów regulujących sposób formowania tabel ustalają style, ale możemy je przedefiniować, np. `\renewcommand{\arraystretch}{1.25}` zmieni odstęp wierszy o czynnik 1,25, albo zmienić prostym przypisaniem typu `\tabcolsep5pt`. Najważniejszymi są:

`\arraycolsep` i `\tabcolsep` — poziomy odstęp tekst – „tabulator” w kolumnie `array` i `tabular`,

`\arrayrulewidth` — grubość linii,

`\doublerulesep` — odstęp między podwójnymi liniami,

`\def\arraystretch{ α }` — α -krotne odstępy wierszy.

■ `\begin{picture}(x,y)(x',y')`

rozkazy graficzne

`\end{picture}`,

gdzie rysunek ma wymiary $x \times y$ z lewym dolnym rogiem w punkcie (x', y') . Polecenia graficzne to m.in.:

`\put(x,y){...}` — wstaw ... w miejscu (x, y)

`\multiput(x,y)(Δx , Δy){n}{...}` — n kopii ... w odstępach Δx w poziomie i Δy w pionie

`\makebox(x,y)[loc]{...}`, gdzie $loc \in t, l, r, c$

`\dashbox{d}(x,y)[loc]{...}`, gdzie d jest długością kresek, z których składa się ramka wokół ...

`\line(h,v){l}` — linia nad podstawą l i o nachyleniu $h \times v$, gdzie $0 \leq h, v \leq 6$

`\vector(h,v){l}` — `\line` z grotem, tj. strzałka ($0 \leq h, v \leq 4$)

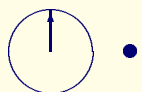
`\circle[*]{d}` — okrąg [dysk] o średnicy d

`\oval(x,y)[part]` — [częściowy] owal; $part \in t, b, r, l, br, \dots$ ($br =$ bottom-right, tj. spód z prawej)

`\frame{...}` — ramka wokół ...

`\shortstack[$\frac{1}{r}$]{...}` — `\vbox{...}`, w którym tekst ... jest składany w jedną kolumnę wyrównaną zgodnie z $[\frac{1}{r}]$, tj. z l lub r jak w *formacie* otoczenia `tabular` i w opcjonalnym argumencie polecenia `\makebox`.

Oto przykład prostego rysunku:



```

\begin{picture}(60,30)(0,0)
  \put(20,0){\circle{30}}
  \put(20,0){\vector(0,1){15}}
  \put(50,0){\circle*{5}}
\end{picture}

```

Poleceniami otoczenia `picture` są m.in. `\thinlines` i `\thicklines` (cienkie i grube linie). Dowolną grubość linii (ale tylko pionowych i poziomych) można zadać poleceniem `\linethickness{dim}`. Wymiary są wyrażone w jednostkach `\unitlength=1pt`, które można zmienić (podobnie jak `\fboxsep=3pt`, `\fboxrule=.4pt`)

Następne dwa otoczenia są ruchome, co oznacza że ich zawartość może być umieszczona przez \LaTeX -a u góry lub dołu strony, albo na osobnej stronie. Polecenia `\caption` (może ich być kilka w jednym otoczeniu) dołączają do rysunku lub tabeli automatycznie numerowany podpis (w wersji oryginalnej z angielskimi słowami **Figure** i **Table**, które zmienimy — w odpowiednim pliku stylu `.sty` — np. na **Rys.** i **Tab.**) W celu scentrowania rysunku lub tabeli możemy na początku danego otoczenia wpisać polecenie `\centering`.

■ `\begin{figure[*]}[loc] ...`

program rysowania, tj. otoczenie picture, albo tylko \vspace{w} w celu zostawienia wolnego miejsca o wysokości w z ewentualnymi

`\caption[Tytuł do spisu rysunków (do 300 liter)] {Podpis rysunku}` i

`\label{etykieta}`

`... \end{figure[*]}`,

gdzie za `loc` można podstawić `h` (od ang. here, czyli tutaj), `t` (top, tj. u góry strony), `b` (bottom, tj. dół) lub/i `p` (cała osobna strona). Wersja `figure*` w formacie (lub z opcją) dwukolumnowym tworzy jednokolumnowe rysunki (tzn. przez całą szerokość strony).

- `\begin{table[*]}[loc] ...`
treść tabeli, tj. np. otoczenie tabular, z ewentualnym
`\caption[Tytuł do spisu tabel (do 300 liter)] {Podpis do tabeli}` i
`\label{etykieta}`
`... \end{table[*]}`,
 gdzie oznaczenia są takie same jak przy `figure`.

Wstawianie ruchomych otoczeń w tekst regulują parametry zależne od stylu. Są to m.in.:

`\floatsep` — odległość między otoczeniami
`\intextsep` — odległość między tekstem i otoczeniem przy opcji `h`
`\textfloatsep` — odległość między tekstem i otoczeniem przy opcjach `t` i `b`
`\topfraction` — ułamek kolumny, jaki może być poświęcony na otoczenie
`\textfraction` — minimalny ułamek kolumny, który musi zawierać tekst
`\floatpagefraction` — minimalny ułamek strony, który musi zająć otoczenie
`\dblfloatpagefraction` — j.w., ale dla dwukolumnowego formatu

12. Etykiety, odwołania i przypisy (*labels, crossreferences i footnotes*)

Mamy trzy podstawowe polecenia do robienia odniesień:

```
\label{nazwa} \ref{nazwa} \pageref{nazwa}
```

W wybranym miejscu tekstu (niechaj po „kompilacji” okaże się to strona 21, rozdz. 2, a punkt 3) piszemy na przykład `\label{odwołanie}`, a w innych miejscach (także wcześniejszych!) możemy odwołać się do niego za pomocą `\sf na stronie \pageref{odwołanie}` albo `\it w punkcie \ref{odwołanie}` uzyskując w pierwszym przypadku efekt „na stronie 21”, zaś w drugim — „w punkcie 2.3,”. Podobnie powołujemy się na numery wzorów matematycznych, w których wcześniej postawiliśmy umowne etykiety. Numery takie autorzy normalnie ujmują w okrągłe nawiasy, zatem powoływalibyśmy się na nie pisząc np. **we wzorze** (`\ref{etykieta}`). Można jednak stworzyć sobie automatyczne ujmowanie w nawiasy definiując nowe polecenie w rodzaju: `\def\mref#1{(\ref{#1})}` i pisząc każdorazowo już tylko: **we wzorze** `\mref{etykieta}`

Przypisy u dołu strony powstają po `\footnote[n] {Tekst przypisu}`⁸. Można też napisać oddzielnie polecenie `\footnotemark` (ewentualnie z `[n]`) w celu postawienia samego

⁸`[n]` można opuścić i wtedy numerowanie odbywa się automatycznie. W tym przypisie za `n` położyliśmy cyfrę 8.

numeru przypisu, a w innym miejscu podać `\footnotetext{Tekst przypisu}` (ewentualnie z `[n]`), co jest w sumie równoważne poleceniu `\footnote{Tekst przypisu}`. Takiego sztukowania wymagają sytuacje, kiedy np. przypis zapragniemy umieścić wewnątrz *boxu*: wewnątrz stawiamy tylko znaczek, a na zewnątrz — tekst.

Istnieje możliwość oznaczania przypisów specjalnymi znaczkami (*, †, ‡, § itd., po kolei 9 znaków), ale wymaga to przedefiniowania polecenia `\thefootnote`. Mianowicie, w naszym dokumencie (albo pliku `.sty`) piszemy:

```
\def\thefootnote{\fnsymbol{footnote}},
```

gdzie `footnote` jest licznikiem przypisów i trzeba go przestawić po osiągnięciu wartości 10, albo używać innego licznika w to miejsce. Zamiast `\fnsymbol{footnote}` piszemy `\alph{footnote}` otrzymując małe litery alfabetu łacińskiego na oznaczenia przypisów (w \LaTeX -u jest `\arabic{footnote}`). Przykładowo, tutaj `footnote` ma wartość 2[‡].

13. Liczniki (*counters*)

Paginator \TeX operuje wieloma licznikami: strony (`page`), sekcje (`chapter`, `section`, `subsection` ...), otoczenia (`figure`, `table`, `enumi`, `enumii`, `enumiii` i `enumiv`), przypisy (`footnote`, `mpfootnote`, ten drugi dotyczy otoczenia `minipage`) czy wzory matematyczne (`equation`).

Licznik można ustawiać:

```
\setcounter{name}{wartosc},
```

inkrementować o `wartosc` lub o 1:

```
\addtocounter{name}{wartosc}, \stepcounter{name},
```

pobrać wartość:

```
\value{name},
```

wypisać w składanym tekście:

```
\thename
```

(np. `\thepage`), wypisać w innym stylu:

```
\arabic{name}, \roman{name}, \Roman{name},
\alph{name}, \Alph{name}
```

(np. w tym miejscu `page` ma wartość 27, a po `\Roman {page}` dostaje się XXVII). Użytkownik może założyć swój licznik: `\newcounter{nazwa}[sect]`, tutaj o nazwie `nazwa`, ustawiony na 0 i „resetowany” na zero po każdej nowej sekcji `sect`.

Istnieją też specjalne liczniki-polecenia daty i czasu: `\year`, `\month`, `\day` i `\time`. Bieżącą datę i czas (w minutach od północy) możemy uzyskać za pomocą poleceń typu

[‡]Wartości tej odpowiada symbol ‡.

`\the\day(=\number\day) --\the\month, \the \year: i \the\time`, które w tym przypadku dały następujący efekt: 22(=22) -1, 2004: i 937.

Dodajmy, że liczniki oparte są na T_EX-owych wewnętrznych rejestrach liczb całkowitych w ilości 256, z czego `\count1, . . . , \count9` przeznaczone są do bezpośredniego korzystania użytkowników. W liczniku 0 (`\count0`) przechowywany jest bieżący numer strony, tak że operacja odejmowania `\advance\count0 by -\value{page}` zeruje obydwie liczniki, o czym możemy przekonać się pisząc np. `\the\count0` i `\thepage`.

14. Znaki diakrytyczne (akcenty)

W wielu językach opartych na alfabecie łacińskim występują pewne szczególne odmiany liter i rozmaite akcenty. Autor T_EX-a przewidział pewną ich liczbę w standardowych zestawach znaków (fontach). M.in. prawie w każdym foncie występuje osobliwa kresciczka (`\char32`) przeznaczona w zasadzie wyłącznie dla polskich liter ł i Ł. Niestety, Knuth nie zapewnił nam równie wyjątkowego ogonka do liter ą, Ą, ę i Ę (te, które Czytelnik tutaj widzi zostały dorobione przez autora niniejszego dokumentu za pomocą METAFONT-a). Jeśli nie dysponujemy specjalnymi „polskimi” fontami to musimy zadowolić się francuską cedyllą: ą, Ȧ, ę i Ẹ, przy czym ten pierwszy przypadek należy zdefiniować raczej tak: `\hbox{\kern.1em\hbox{\char24} □\kern-.9em□a}`, gdyż `\c a` daje ą.

W tabelce przedstawiamy podsumowanie standardowych akcentów L^AT_EX-a. Zwróćmy uwagę na inny sposób akcentowania w trybie matematycznym niż tekstowym. Akcenty tekstowe możemy wprowadzić do wyrażeń trybu *math* używając *hboxów*, np. `$\hbox{\it \b x}$` daje *x*. W ogólności, akcenty tekstowe możemy postawić za pomocą polecenia `\accentN`, gdzie N jest numerem znaku w foncie branego jako akcent, np. `\accent39x` daje *ř*. W trybie *math* używa się poleceń typu `\mathaccent"7012` (co w tym wypadku jest równoważne `\grave`).

	Tekstowe			Matematyczne
19	<code>\'m</code>	acute	<i>m</i>	<code>\acute m</code>
18	<code>\'m</code>	grave	<i>m</i>	<code>\grave m</code>
94	<code>\^m</code>	hat	<i>m</i>	<code>\hat m</code>
126	<code>\~m</code>	tilde	<i>m</i>	<code>\tilde m</code>
127	<code>\"m</code>	ddot	<i>m</i>	<code>\ddot m</code>
22	<code>\=m</code>	bar	<i>m</i>	<code>\bar m</code>
95	<code>\.m</code>	dot	<i>m</i>	<code>\dot m</code>
20	<code>\v m</code>	check	<i>m</i>	<code>\check m</code>
21	<code>\u m</code>	breve	<i>m</i>	<code>\breve m</code>
<hr/>				
125	<code>\H m</code>	vec	<i>m</i>	<code>\vec m</code>
24	<code>\c m</code>	widetilde	<i>m</i>	<code>\widetilde m</code>
	<code>\t nn</code>	widetilde	<i>mno</i>	<code>\widetilde{mno}</code>
	<code>\b m</code>	widehat	<i>m</i>	<code>\widehat m</code>

15. Wyrażenia matematyczne

Praca w trybie matematycznym (*math*) jest szczególnie przyjemna ze względu na łatwość osiągnięcia pięknych składów. Trzeba pamiętać, że symbole matematyczne i fizyczne drukowane w tekście powinny mieć inny krój niż tekst — zwykle jest to italika, a w T_EX-u „matematyczna italika” uzyskiwana przez otoczenie symboli znakami \$. Np. $\$x\$$ daje *x*. Tak samo piszemy całe wyrażenia matematyczne, ale można też inaczej, gdyż równoważne są:

```

$ x\over y + z $
\(\x\over y +z\)
\begin{math} x\over y+z \end{math}

```

— wszystkie dają efekt $\frac{x}{y+z}$. Zatem jedną z tych trzech form użyjemy pisząc wzór w tekście.

Jeśli chcemy wzór wyróżnić z tekstu, to mamy znów do dyspozycji kilka możliwości:

```

$$ {x \over y} = z^2 $$
\[\quad {x\over y} = z^2 \quad \]
\begin{displaymath} { x \over y}=z^2
\end{displaymath}
\begin{equation} {x\over y}=z^2 \quad \nonumber
\end{equation}
\begin{eqnarray*} {x \over y} = z^2
\end{eqnarray*}

```

— wszystkie dają podobny efekt:

$$\frac{x}{y} = z^2$$

Wreszcie, równania numerowane z prawej strony dostaniemy używając otoczeń `equation` i `eqnarray`, przy czym to drugie służy do składania układów równań. Używając jednej z możliwości:

```

\begin{equation}\sqrt{x} = z_2{a\over b}
\end{equation}
\begin{eqnarray}\sqrt{x} = z_2{a\over b}
\end{eqnarray}

```

dostaje się

$$\sqrt{x} = z_2 \frac{a}{b} \tag{1}$$

(numerek otrzymalibyśmy z lewej strony, gdyby ten dokument miał opcję `leqno` przy `\documentstyle`; nie należy mylić jej z poleceniem `\leqno`, którego użyjemy niżej)

Oto przykład układu równań, z których pierwsze pozbawiliśmy numeru poleceniem `\nonumber`:

$$\begin{aligned} x^2 &= ay + bx \\ y^2 &= cy - dx \end{aligned} \tag{2}$$

a w pliku źródłowym odpowiada mu zapis

```
\begin{eqnarray} x^2&=&ay+b x \nonumber\\ y^2&=&c y-dx \end{eqnarray}
```

Polecenie `\setcounter{equation}{12}` ustawia bieżący numer równania. Można go przypisać także „na piechotę” wyrażeniu normalnie nienumerowanemu. Np.,

```
$$\{\cal A}\buildrel\rm def\over= {\partial\over\partial a}
```

(12a)
$$\mathcal{A} \stackrel{\text{def}}{=} \frac{\partial}{\partial a}$$

albo inaczej (i nieco sztucznie): `$$x^2-y^2=(x-y)(x+y) \eqno (7-b,\theequation)$$`

$$x^2 - y^2 = (x - y)(x + y) \quad (7 - b, 12)$$

Po `\lefteqn{...}` T_EX traktuje równanie tak, jak gdyby miało zerową szerokość, dzięki czemu możemy lepiej rozmieścić długie wzory, np.

```
\begin{eqnarray*} \lefteqn{x+y+z+t=a+b\ \null}\\ &\null+c+d+e+f+g+h-\arcsin(\cos\alpha)-\ln 5-\\ &\null-i+j+k-l \end{eqnarray*}
```

daje wynik

$$\begin{aligned} x + y + z + t &= a + b + \\ &+ c + d + e + f + g + h - \arcsin(\cos \alpha) - \ln 5 - \\ &- i + j + k - l \end{aligned}$$

Zwróćmy uwagę na obecność `\null` (pusty *hbox*). T_EX prawidłowo rozróżnia sytuacje `$-x$` (tj. $-x$) i `$a-x$` (tj. $a - x$).

Polecenie `*` w trybie *math* zachowuje się tak, jak `\-` w trybie tekstowym, tj. wskazuje dodatkowe miejsce podziału wzoru między linie; skuteczny podział jest oznaczany przez `\allowbreak` wskazuje takie miejsca podziału bez oznaczania.

W trybie matematycznym polski przecinek dziesiętny (inaczej niż kropka) wymaga korekty; porównaj np. 3,14 3,14 i 3.14 uzyskane za pomocą `$3,14$` `$3, \!14$` i `3.14`.

Zauważyliśmy zapewne, że spacje (odstęp) w omawianym trybie są generalnie ignorowane (oczywiście, w razie potrzeby, możemy je wymusić na kilka sposobów).

Oto dalsze pouczające przykłady — wszystkie w stylu tekstowym, tj. typu `$. . . $`.

```
 $\frac{2}{1}x^4, x^{2y}$  ← $_1^2x_3^4, \ x^{\{2^y\}}$  

 $\overline{\underline{z}}$  ← $ \overline{\underline{z}}$  

 $A \overset{a'}{\rightarrow} B.$  ← $A\stackrel{a'}{\rightarrow} B.$  

 $\dot{A} dx dy dz$  ← $\skew6\dot{\vec A}\ ;dx\>dy\ ,dz$4  

 $((x^2)^3)^4 ((x^2)^3)^4$  ← $((x^2)^3)^4 \ {((x^2)^3)}^4$  

 $y^{\frac{2}{k+1}} + z^{2/(k+1)}$  ← $y^{\{2\over k+1\}}+z^{\{2/(k+1)\}}$
```

⁴Składnia polecenia użytego w tym przykładzie jest następująca: `\skew{h}{gora}{dol}`, gdzie wysokość *h* wyrażona jest w jednostkach mu. $1\mu = \frac{1}{18}$ em (stąd mamy polecenia `\thinmuskip = 3mu` albo `\,`, `\medmuskip = 4mu` plus `2mu` minus `4mu` albo `\>`, `\thickmuskip=5mu` plus `5mu` albo `\;`);

$y^{\frac{2}{k+1}}$	\leftarrow	<code>\textstyle{2\over k+1}</code>	<code>\\$y^{\textstyle{2\over k+1}}\\$</code>
$z^{2/k}$	\leftarrow	<code>\displaystyle{2/k}</code>	<code>\\$z^{\displaystyle{2/k}}\\$</code>
$y_3''', y_3''''^2$	\leftarrow	<code>\prime\prime\prime</code> , <code>\prime\prime\prime^2_3</code>	<code>\\$y_3^{\prime\prime\prime}, y_3^{\prime\prime\prime^2}\\$</code>
$\sqrt[n]{x^2 - \overline{y}}$	\leftarrow	<code>\root n \of{x^2-\overline{y}}</code>	<code>\\$\root n \of{x^2-\overline{y}}\\$</code>
$\binom{n+1}{3} \left[\begin{matrix} n+2 \\ m \end{matrix} \right]$	\leftarrow	<code>\{n+1\choose 3} \{n+2\brack m}</code>	<code>\\$\{n+1\choose 3} \{n+2\brack m}\\$</code>
$\left\{ \begin{matrix} m+7 \\ k-l-m+x \end{matrix} \right\}$	\leftarrow	<code>\{m+7\brace k-l-m+x}</code>	<code>\\$\{m+7\brace k-l-m+x}\\$</code>
$\binom{a}{b}$	\leftarrow	<code>\{a \atopwithdelims \lgroup \rgroup b}</code>	<code>\\$\{a \atopwithdelims \lgroup \rgroup b}\\$⁵</code>
$\left\langle \begin{matrix} a \\ b \end{matrix} \right\rangle$	\leftarrow	<code>\\$a\abovewithdelims<>3pt b\\$</code>	<code>\\$a\abovewithdelims<>3pt b\\$</code>
$\left \frac{a}{b} \right , n \bmod m$	\leftarrow	<code>\{a\overwithdelims b}, \ n\bmod m</code>	<code>\\$\{a\overwithdelims b}, \ n\bmod m\\$</code>
$\frac{a}{b} \quad \frac{k}{l}$	\leftarrow	<code>\{a \above 2pt b}\quad\{k\atop l}</code>	<code>\\$\{a \above 2pt b}\quad\{k\atop l}\\$</code>
$\sum_{n=1}^3 z_n^2$	\leftarrow	<code>\sum\limits_{n=1}^3 z_n^2</code>	<code>\\$\sum\limits_{n=1}^3 z_n^2\\$</code>
$\sum_{n=2}^6 a_i$	\leftarrow	<code>\sum_{n=2}^6 a_i</code>	<code>\\$\sum_{n=2}^6 a_i\\$</code>
$\sum_{\substack{0 \leq i < m \\ 0 \leq j < n}}$	\leftarrow	<code>\sum\limits_{\scriptstyle 0 \leq i < m \\ \scriptstyle 0 \leq j < n}</code>	<code>\\$\sum\limits_{\scriptstyle 0 \leq i < m \\ \scriptstyle 0 \leq j < n}</code>
$\int_{-\infty}^{\pi/2} y \sin x$	\leftarrow	<code>\int\limits_{-\infty}^{\pi/2} y \sin x</code>	<code>\\$\int\limits_{-\infty}^{\pi/2} y \sin x\\$</code>
$\lim_{k \rightarrow -\infty} y^k$	\leftarrow	<code>\lim\limits_{k \rightarrow -\infty} y^k</code>	<code>\\$\lim\limits_{k \rightarrow -\infty} y^k\\$</code>
$\underbrace{x+y+z}_{\leq 0}$	\leftarrow	<code>\underbrace{x+y+z}_{\leq 0}</code>	<code>\\$\underbrace{x+y+z}_{\leq 0}\\$</code>
$a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3}}}$	\leftarrow	<code>\\$a_0 + \{1\over\displaystyle a_1 + \{\strut 1\over \displaystyle a_2 + \{\strut \frac{1}{a_3}\}\}\}\\$</code>	<code>\\$a_0 + \{1\over\displaystyle a_1 + \{\strut 1\over \displaystyle a_2 + \{\strut \frac{1}{a_3}\}\}\}\\$</code>
$ x = \begin{cases} x, & \text{gdy } x \geq 0; \\ -x, & \text{gdy } x < 0. \end{cases}$	\leftarrow	<code>\\$ x = \cases{x, & \text{gdy } x \geq 0; \\ -x, & \text{gdy } x < 0. \end{cases}\\$</code>	<code>\\$ x = \cases{x, & \text{gdy } x \geq 0; \\ -x, & \text{gdy } x < 0. \end{cases}\\$</code>
$A = \begin{pmatrix} x-l & 1 & 0 \\ 0 & x-l & 1 \\ 0 & 0 & x-l \end{pmatrix}$	\leftarrow	<code>\\$A = \matrix{x-l&1&0 \\ \cr 0&x-l&1 \\ \cr 0&0&x-l}\\$</code>	<code>\\$A = \matrix{x-l&1&0 \\ \cr 0&x-l&1 \\ \cr 0&0&x-l}\\$</code>

Taką samą macierz uzyskamy po `\$A = \left(\matrix{x-l& 1&0 \\ \cr 0&x-l&1 \\ \cr 0&0&x-l}\right)\$`

Sposób użycia otoczenia `array` (opis patrz s. 23) ilustruje następujący przykład (celowo zostawiliśmy pewne niedociągnięcia redakcyjne):

```

 $\vec{x} = \left( \begin{array}{rcc}
x_1 & = & \log_{10}(\sin^3 y) \equiv 3 \log \sin y \\
x_2 & = & \operatorname{Im}(z) + \operatorname{Re}(z) + e^y - e^{-y} \\
& & \vdots \\
x_{10} & = & \begin{array}{l} 1, \text{ dla } j=2k \\ 3, \text{ dla } j=2k+1 \end{array} \end{array} \right)$ 

```

⁵Polecenie użyte nieco wyżej, `\brace`, zostało zdefiniowane tak:
`\def\brace{\atopwithdelims\{}`

```

4,&\hbox{dla} j=-5\\
\end{array}\right)\}
3,&\hbox{gdy } & t>kjb\\
0,&\hbox{ gdy} & t<0\\
-1,& \hbox{gdy }& t\ll\Xi\\
\end{array}
\end{array}\right. $$

```

który po przetworzeniu daje wydruk:

$$\vec{x} = \left\{ \begin{array}{l} x_1 = \log_{10}(\sin^3 y) \equiv 3 \log \sin y \\ x_2 = \Im(z) + \Re(z) + e^y - e^{-y} \\ \vdots \\ 1, \text{ gdy } t = \begin{cases} 3, & \text{dla } j = 2k \\ 4, & \text{dla } j = -5 \end{cases} \\ 3, \text{ gdy } t > kjb \\ 0, \text{ gdy } t < 0 \\ x_{10} = -1, \text{ gdy } t \ll \Xi \end{array} \right.$$

`\leftarrowfill` zostało użyte wcześniej w tym punkcie dla lewostronnej strzałki wypełniającej. Podobnie działają `\dotfill`, `\rightarrowfill` i `\hrulefill`. Np., tutaj po `\dotfill\par` uzyskaliśmy:

Funkcje

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

Znak + w tej tabelce wskazuje, że funkcja ma dolne i górne wskaźniki traktowane jako „granice”.

Litery greckie

α	<code>\alpha</code>	ϖ	<code>\varpi</code>	Π	<code>\Pi</code>
β	<code>\beta</code>	ρ	<code>\rho</code>	Σ	<code>\Sigma</code>
γ	<code>\gamma</code>	ϱ	<code>\varrho</code>	Υ	<code>\Upsilon</code>
δ	<code>\delta</code>	σ	<code>\sigma</code>	Φ	<code>\Phi</code>
ϵ	<code>\epsilon</code>	ς	<code>\varsigma</code>	Ψ	<code>\Psi</code>
ε	<code>\varepsilon</code>	τ	<code>\tau</code>	Ω	<code>\Omega</code>
ζ	<code>\zeta</code>	υ	<code>\upsilon</code>	Γ	<code>\mit\Gamma</code>
η	<code>\eta</code>	ϕ	<code>\phi</code>	Δ	<code>\mit\Delta</code>
θ	<code>\theta</code>	φ	<code>\varphi</code>	Θ	<code>\mit\Theta</code>

ϑ	<code>\vartheta</code>	χ	<code>\chi</code>	Λ	<code>\mit\Lambda</code>
ι	<code>\iota</code>	ψ	<code>\psi</code>	Ξ	<code>\mit\Xi</code>
κ	<code>\kappa</code>	ω	<code>\omega</code>	Π	<code>\mit\Pi</code>
λ	<code>\lambda</code>	Γ	<code>\Gamma</code>	Σ	<code>\mit\Sigma</code>
μ	<code>\mu</code>	Δ	<code>\Delta</code>	Υ	<code>\mit\Upsilon</code>
ν	<code>\nu</code>	Θ	<code>\Theta</code>	Φ	<code>\mit\Phi</code>
ξ	<code>\xi</code>	Λ	<code>\Lambda</code>	Ψ	<code>\mit\Psi</code>
π	<code>\pi</code>	Ξ	<code>\Xi</code>	Ω	<code>\mit\Omega</code>

Małe litery greckie traktowane są przez TeX-a jako symbole matematyczne, zaś duże — jako litery. `gg\Gamma,gg\gamma` otrzymamy po `\boldmath $\rm g\bf g\gamma\Gamma\Gamma, \mit g\gamma\Gamma\Gamma$}`. Polecenie odwracające `\boldmath`, to `\unboldmath`.

Operatory, relacje i inne symbole

\pm	<code>\pm</code>	\cap	<code>\cap</code>	\vee	<code>\vee</code>
\mp	<code>\mp</code>	\cup	<code>\cup</code>	\wedge	<code>\wedge</code>
\setminus	<code>\setminus</code>	\uplus	<code>\uplus</code>	\oplus	<code>\oplus</code>
\cdot	<code>\cdot</code>	\sqcap	<code>\sqcap</code>	\ominus	<code>\ominus</code>
\times	<code>\times</code>	\sqcup	<code>\sqcup</code>	\otimes	<code>\otimes</code>
$*$	<code>\ast</code>	\triangleleft	<code>\triangleleft</code>	\oslash	<code>\oslash</code>
\star	<code>\star</code>	\triangleright	<code>\triangleright</code>	\odot	<code>\odot</code>
\diamond	<code>\diamond</code>	\wr	<code>\wr</code>	\dagger	<code>\dagger</code>
\circ	<code>\circ</code>	\bigcirc	<code>\bigcirc</code>	\ddagger	<code>\ddagger</code>
\bullet	<code>\bullet</code>	\bigtriangleup	<code>\bigtriangleup</code>	\amalg	<code>\amalg</code>
\div	<code>\div</code>	\bigtriangledown	<code>\bigtriangledown</code>	\mho	<code>\mho</code>
		\square	<code>\Box</code>	\diamond	<code>\Diamond</code>
\leadsto	<code>\leadsto</code>	\triangleleft	<code>\lhd</code>	\triangleleft	<code>\unlhd</code>
$\not\leadsto$	<code>\not\leadsto</code>	\triangleright	<code>\rhd</code>	\triangleright	<code>\unrhd</code>
\leq	<code>\leq</code>	\geq	<code>\geq</code>	\equiv	<code>\equiv</code>
\prec	<code>\prec</code>	\succ	<code>\succ</code>	\sim	<code>\sim</code>
\preceq	<code>\preceq</code>	\succeq	<code>\succeq</code>	\simeq	<code>\simeq</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\asymp	<code>\asymp</code>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\sqsubseteq	<code>\sqsubseteq</code>	\cong	<code>\cong</code>
\supseteq	<code>\supseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\Join	<code>\Join</code>
\in	<code>\in</code>	\ni	<code>\owns</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
\smile	<code>\smile</code>	\mid	<code>\mid</code>	\doteq	<code>\doteq</code>
\frown	<code>\frown</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
\neq	<code>\neq</code>	\notin	<code>\notin</code>	\parallel	<code>\parallel</code>
\aleph	<code>\aleph</code>	\prime	<code>\prime</code>	\forall	<code>\forall</code>
\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>
\imath	<code>\imath</code>	∇	<code>\nabla</code>	\neg	<code>\neg</code>
\jmath	<code>\jmath</code>	\triangle	<code>\triangle</code>	\flat	<code>\flat</code>
ℓ	<code>\ell</code>	∂	<code>\partial</code>	\natural	<code>\natural</code>

\wp	<code>\wp</code>	\int	<code>\smallint</code>	\sharp	<code>\sharp</code>
\Re	<code>\Re</code>	\clubsuit	<code>\clubsuit</code>	\parallel	<code>\Vert</code>
\Im	<code>\Im</code>	\diamond	<code>\diamondsuit</code>	\angle	<code>\angle</code>
\top	<code>\top</code>	\heartsuit	<code>\heartsuit</code>	\surd	<code>\surd</code>
\perp	<code>\bot</code>	\spadesuit	<code>\spadesuit</code>	\cdots	<code>\cdots</code>
∞	<code>\infty</code>	\backslash	<code>\backslash</code>	\dots	<code>\ldots</code>
\vdots	<code>\vdots</code>	\ddots	<code>\ddots</code>	\dots	<code>\dotfill</code>

Polecenia `\mid` i `\parallel` są takie same jak `|` i `||`, ale są otoczone większymi odstępami. `\prime` używa się głównie do wskaźników. Pewne dodatkowe symbole można łatwo skonstruować z istniejących, np.

`\not\subset` \neq `\not=` $\hat{}$ `\hat\imath`

Strzałki i operatory

\leftarrow	<code>\leftarrow</code>	<code>\gets</code>	\longleftarrow	<code>\longleftarrow</code>
\Lleftarrow	<code>\Lleftarrow</code>		\Longleftarrow	<code>\Longleftarrow</code>
\rightarrow	<code>\rightarrow</code>	<code>\to</code>	\longrightarrow	<code>\longrightarrow</code>
\Rrightarrow	<code>\Rrightarrow</code>		\Longrightarrow	<code>\Longrightarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>		\longleftrightarrow	<code>\longleftrightarrow</code>
\Lleftrightarrow	<code>\Lleftrightarrow</code>		\Longleftrightarrow	<code>\Longleftrightarrow</code>
\uparrow	<code>\uparrow</code>		\Uparrow	<code>\Uparrow</code>
\downarrow	<code>\downarrow</code>		\Downarrow	<code>\Downarrow</code>
\updownarrow	<code>\updownarrow</code>		\Updownarrow	<code>\Updownarrow</code>
\nearrow	<code>\nearrow</code>		\searrow	<code>\searrow</code>
\swarrow	<code>\swarrow</code>		\nwarrow	<code>\nwarrow</code>
\mapsto	<code>\mapsto</code>		\longmapsto	<code>\longmapsto</code>
\hookrightarrow	<code>\hookrightarrow</code>		\hookrightarrow	<code>\hookrightarrow</code>
\leftharpoonup	<code>\leftharpoonup</code>		\leftharpoonup	<code>\leftharpoonup</code>
\rightharpoonup	<code>\rightharpoonup</code>		\rightharpoonup	<code>\rightharpoonup</code>
\rightharpoons	<code>\rightharpoons</code>		\iff	<code>\iff</code>
Σ	<code>\sum</code>		\bigcap	<code>\bigcap</code>
\prod	<code>\prod</code>		\bigcup	<code>\bigcup</code>
\coprod	<code>\coprod</code>		\bigsqcup	<code>\bigsqcup</code>
\bigvee	<code>\bigvee</code>		\bigoplus	<code>\bigoplus</code>
\bigwedge	<code>\bigwedge</code>		\bigotimes	<code>\bigotimes</code>
\bigodot	<code>\bigodot</code>		\bigoplus	<code>\bigoplus</code>
\int	<code>\int</code>		\oint	<code>\oint</code>

`\iff` jest takie, jak `\Longleftrightarrow`, ale z większymi odstępami po bokach.

Operatory występują w dwóch rozmiarach, z których ten większy objawia się przy `\displaystyle`.

Ograniczniki

(([<code>\lbrack</code>	[{	<code>\lbrace</code>
))]	<code>\rbrack</code>]	}	<code>\rbrace</code>
[<code>\lfloor</code>	⌈	<code>\lceil</code>	<	<code>\langle</code>	<
]	<code>\rfloor</code>	⌋	<code>\rceil</code>	>	<code>\rangle</code>	>
	<code>\vert</code>		<code>\ </code>	\	<code>\backslash</code>	
/	/					

Wszystkie strzałki w górę lub w dół są także ogranicznikami i takim jest również specjalna forma „.” służąca za „pusty” ogranicznik dopełniający parę z `\left` lub `\right`. Znaki `<` i `>` można używać zamiast `\langle` i `\rangle` po `\left` i `\right`.

Przed delimiterami (ogranicznikami) możemy napisać polecenia powiększające je nieznacznie (`\big`) i jeszcze o czynnik 1,5 (`\Big`), albo 2,0 (`\bigg`), albo 2,5 (`\Bigg`). Nazwy wymienionych poleceń możemy ponadto rozszerzyć o literę `l`, `m` bądź `r` (np. `\bigl`) przypisując następującym po nich znakom własności lewego ogranicznika, relacji bądź prawego ogranicznika, odpowiednio. Przykładowo, `\left \arrowvert \right \Arrowvert \Bigg \Arrowvert` daje $\left\| \right\|$. Zapamiętajmy także, że delimitery zmieniają swe rozmiary po `\left` i `\right`.

16. Definicje i makrodefinicje

W plikach źródłowych `plain.tex` bądź `lplain.tex` (albo `mlplain.tex` bądź `mllplain.tex`, które są modyfikacjami wielojęzycznego T_EX-a) i `latex.tex` polecenia definiowane są instrukcją `\def\pojeczenie{Tre/s/c}`. Napisanie w tekście `\pojeczenie` jest równoważne `Tre/s/c` (ale, uwaga, bez nawiasów klamrowych!). W `plain.tex` mamy `m.in`.

```
\def\space{ } \def\empty{} \def\null{\hbox{}}.
```

Definicje mogą zawierać do 9–ciu argumentów: `#1`, `#2`, ..., `#9`). Np.

```
\def\wektor#1#2{(#1_1,\ldots,#1_{#2})}
```

po `\$ \wektor ynp $`, albo po `\$ \wektor{y}{n}p $`, daje efekt: $(y_1, \dots, y_n)p$, zaś po `\$ \wektor x{klm}q $` — $(x_1, \dots, x_{klm})q$. Jeśli wewnątrz definicji występuje inna definicja, to jej ewentualne argumenty oznacza się przez `##1`, `##2`, Gdy w definicji po argumentie formalnym wystąpi znak różny od `#` i `{`, to przy wywoływaniu takiego polecenia za odpowiedni argument podstawiany jest cały tekst aż do napotkania tego znaku (lub ciągu znaków). Np. dla przypadku `\def\test#1ABC#2.#3${...}`, po wystąpieniu w tekście źródłowym polecenia `\test`, pierwszy argument (`#1`) otrzyma wszystkie znaki do najbliższego `ABC`, za drugi podstawiony będzie dalszy fragment do kropki, a za trzeci — kolejny fragment zakończony znakiem `$`.

Definicja umieszczona wewnątrz nawiasów klamrowych (albo wewnątrz innej definicji) jest skuteczna tylko w ich obrębie. Gdy chcemy ją udostępnić na zewnątrz, to poprzedzamy ją poleceniem `\global` (albo zamiast `\def` piszemy `\gdef`). Przypisanie istniejącej definicji innej nazwie zapewnia konstrukcja typu `\let\pojeczenie=\wektor`, co powoduje efekt taki, jak skopiowanie definicji, w tym przypadku `\wektora` na `\pojeczenie`.

Komenda `\edef` działa podobnie jak `\def`, ale jest rozwijana przy wczytywaniu, a nie dopiero przy użyciu. Obrazuje to następujący przykład:

```
\def\inf{Stare} \edef\einf{\inf}
\def\test{\inf} \def\inf{Nowe}
```

Sekwencja `\inf\test\einf` daje `NoweNoweStare`. Niekiedy makrodefinicja zawiera polecenia, które nie powinny być rozwijane (np. mogą być jeszcze nie określone). Wówczas trzeba je poprzedzić poleceniem `\noexpand`. Np.

```
\def\inf{Stare} \edef\einf{\inf---\noexpand\inf}
\def\inf{Nowe}
```

po `\inf\einf` daje `NoweStare—Nowe`. Wersją globalną `\edef` jest `\xdef` (albo `\global\edef`).

Argumenty rzeczywiste normalnie nie mogą zawierać więcej niż jeden paragraf. Polecenie `\long` przed definicją pozwala używać argumentów dłuższych.

Warunkowe wykonanie ciągów poleceń umożliwia konstrukcja:

```
\if warunek gdy prawda \else gdy fałsz \fi
```

gdzie ciąg **gdy prawda/fałsz** jest ewaluowany wtedy, gdy *warunek* jest spełniony/niespełniony. Zamiast `\if warunek` można korzystać z form `\ifnum` i `\ifdim` z relacjami `<, = i >`, np.

```
\ifnum 17>\count0 ... \else ... \fi
\ifdim \leftskip<3cm ... \else ... \fi.
```

Prócz `\if`, sprawdzającego równość dwóch obiektów, dostępne są też formy:

```
\ifodd — sprawdzający czy następująca liczba jest nieparzysta
\ifcat — czy kategorie6 dwóch znaków są takie same
\ifx — czy dwa polecenia rozwijają się w to samo
\ifvmode — czy jest tryb wertykalny (między paragrafami lub w \vbox-ie)
\ifhmode — czy jest tryb horyzontalny (wewnątrz paragrafu lub \hbox-u)
\ifmmode — czy jest tryb matematyczny
\ifinner — czy wewnątrz \hbox/\vbox-u
\ifhbox/\ifvbox N — czy box N zawiera \hbox/\vbox
\ifvoid N — czy box N jest pusty
```

`\ifcase n ... (0) \or ... (1) \or ... (2) \else ... \fi` — ewaluowana jest sekwencja `... (0)`, `... (1)`, `... (2)`, `.....` lub `...`, gdy *n* ma wartość 0, 1, 2, ... lub inną, odpowiednio.

Do już bogatych możliwości definicyjnych T_EX-a L^AT_EX dodaje jeszcze kilka wygodnych poleceń pozwalających na łatwe definiowanie nawet bardzo złożonych konstrukcji. Dobrym przykładem jest

●● `\newenvironment{name}[n]{pocz}{koniec}`

— nowe otoczenie o nazwie `name`, `n` argumentach oraz poleceniach otwierających `pocz`

⁶Każdy znak jest przypisany do jednej z kilkunastu kategorii: kategoria 0 — `\`, 1 — `{`, 2 — `}`, 3 — `$`, 4 — `&`, 5 — `<CR>`, 6 — `#`, 7 — `^`, 8 — `_`, 9 — `<Null>`, 10 — `<Tab>`, 11 — wszystkie litery, 12 — inne znaki, 13 — znaki „aktywne” (np. `~`), 14 — `%`, 15 — nieważne znaki (np. ``).

(ewentualnie z argumentami w postaci `{#1} {#2} ... {#n}`) i zamykających `koniec`. Użycie w ten sposób zdefiniowanego otoczenia wygląda tak:

```
\begin{name}{arg1}{arg2} ... {argn}
zawartość otoczenia
\end{name}
```

W tekście spowoduje to podstawienie przez \TeX -a za `\begin{name}` poleceń bądź tekstu wpisanego w definicji w miejscu pocz, przetworzenie obszaru *zawartość otoczenia* i wykonanie poleceń nazwanych tu `koniec` (może być to zbiór pusty). Analogicznie zmienia się istniejące otoczenia:

```
\renewenvironment{name}[n]{beg}{end}.
```

Podobnie definiuje się też nowe, specyficznie \LaTeX -owe polecenia:

```
\newcommand{name}[n]{...},
```

z analogiczną redefinicją `\renewcommand`. W prostszym przypadku (bez argumentów) definicja taka jest to równoważna poznanej już \TeX -owj `\def\name{...}`.

I jeszcze jedna możliwość w \LaTeX -u; konstrukcja

●● `\newtheorem{name}{Twierdz.}[sect]`

służy do zdefiniowania nowego numerowanego otoczenia typu twierdzenie, o nazwie `name`, oznaczanego etykietą **Twierdz.** i kolejnym numerem [opcjonalnie numerem w sekcji `sect`, gdzie `sect` może być np. `chapter`; jeśli ten argument opuścimy, to numeracja odbywa się niezależnie od sekcji]. Na przykład,

```
\newtheorem{twierdzenie}{Twierdzenie} % musi
% wyst/api/c przed pierwszym u/zyciem
\begin{twierdzenie}
Do standardowych system/ow typu\ / {\em desktop
publishing} \TeX\ ma si/e tak, jak ekspert do
przeci/etnych.
\end{twierdzenie}
```

umieszczone tutaj bezpośrednio po dwukropku daje efekt:

Twierdzenie 1 *Do standardowych systemów typu desktop publishing \TeX ma się tak, jak ekspert do przeciętnych.*

Tutaj jest słowo podniesione o 3pt, teraz fragment opuszczony i ponownie o tyle samo podniesiony; przesunięcia takie uzyskuje się poleceniami `\raise3pt \hbox{...}`, `\lower3pt \hbox{...}` oraz `\lower-3pt \hbox{...}`

Istnieją też całkowicie analogiczne polecenia (`\moveleft`, `\moveright`), które działają w poziomie. Np. sekwencja `\vbox {\hbox{AAAAA}\moveleft8pt\hbox{BBBBB
AAAAA
BBBBB
}\hbox {CCCCC.}}` daje następujący efekt: CCCCC. (Warto zwrócić uwagę na umieszczenie `\vbox`-u względem wiersza, w którym występuje. Inaczej to wygląda dla `\vtop`.)

17. Kilka przykładów makrodefinicji

Aby wysunąć znak poza lewy margines tekstu w danym wierszu (np. w celu jego zaznaczenia) Knuth (1990) poleca przygotować kilka definicji:

```
\def\strutd{\dp\strutbox}
\def\Mzn#1{\strut\vadjust{\kern-\strutd\mzn#1}}
\def\mzn#1{\vtop to \strutd{\baselineskip
\strutd\vss\llap{#1}\null}}
```

Wybrany znak poprzedzamy teraz poleceniem `\Mzn`, np. `\Mzn`, uzyskując efekt taki, jak w poprzednim wierszu.

Oto przykład definiowania nowego operatora do trybu matematycznego (Krieger i Schwarz, 1990):

```
\def\squarebox#1{\mathop{\mkern0.5\thinmuskip
\vbox{\hrule\hbox{\vrule\hskip#1\vrule height#1
width 0pt\vrule}\hrule}\mkern0.5\thinmuskip}}
```

Pisząc `$$\squarebox{7pt}_{i=1}^n\Gamma_i=B.$$` otrzymujemy

$$\prod_{i=1}^n \Gamma_i = B.$$

Wielką literę jako namiastkę zdobnego inicjału spotykanego w dawnych książkach na początkach akapitów możemy wstawiać do tekstu pisząc `\inic` przed pierwszą literą akapitu, jeśli przygotujemy wcześniej makrodefinicję podobną do tej:

```
\def\inic#1{\setbox1\hbox to 27pt {\Huge#1\hfil}
\setbox1\hbox{\lower\ht1\hbox{\raise1.6ex\box1}}
\dp1=0pt\hangindent27pt\hangafter-2
\noindent\hskip-27pt\box1}
```


Boxy i linie można złożyć tak, by obramować tekst. Jednym ze sposobów (Doob, 1992) jest wstawienie obramowywanego tekstu w *hbox* lub *vbox* otaczając go obustronnie pionową linią (`\vrule`). Następnie wstawiamy naszą konstrukcję do *vboxu* rysując poziome linie nad i pod nią:

```
\vbox{\hrule
      \hbox{\vrule\ Nasz tekst \vrule}
      \hrule}
```

Otrzymamy `\Nasz tekst`, co nie prezentuje się nadto estetycznie. Jeśli jednak `Nasz tekst` poprzedzimy `rozporą \strut`, to dostaniemy znacznie lepszy efekt: `\Nasz tekst`.

Na podobnych zasadach opierają się bardziej wyszukane makrodefinicje ramek. Przytoczymy kilka, głównie za Kriegerem i Schwarzem (1990). Obramowanie ze s. 38 uzyskaliśmy definiując makrokomendę:

```
\long\def\Frame#1#2#3{% #1 odlegl. ramka - tekst
%                          #2 grubosc linii ramki
%                          #3 tekst wewnatrz ramki
  \vbox{\hrule height#2 \hbox{\vrule width#2
  \hskip#1 \vbox{\vskip#1{ }#3\vskip#1}\hskip#1
  \vrule width#2} \hrule height#2}}
```

Dwukrotne wywołanie `\Frame` dało pożądaną efekt:
`$$\Frame{1mm}{2.pt}{\Frame{5mm}{.4pt}{\hspace =.75\hspace Tutaj jest s/lowo
\raise3 pt \hbox{podniesione} o 3pt, ... }}$$`.

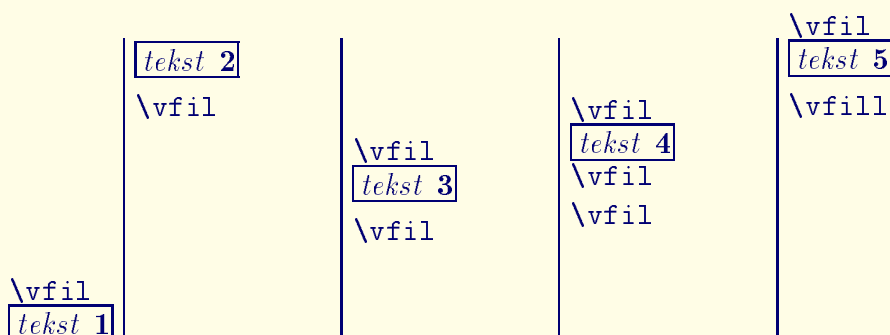
Prostszą konstrukcję ma podobna definicja wykorzystana w następnej ramce:

```
\long\def
\ramka#1#2{\vbox{\hrule\hbox{\vrule\hskip#1\vbox
  {\vskip#1{ }#2\vskip#1}\hskip#1\vrule}\hrule}}
```

gdzie argumentami są odległość ramki od tekstu (`#1`) i sam obramowywany tekst (`#2`).

Dynamiczne odstępy

Jeżeli tekst w `\vbox`-ie o określonej wysokości otoczymy obustronnie poleceniami `\vfil` i/lub `\vfill`, to ów tekst zostanie umieszczony na wysokości odpowiedniej do tych poleceń. Oto przykład dla pięciu takich samych `\vbox`-ów, o wysokościach 4 cm, w których umieściliśmy teksty objęte tutaj małymi ramkami i otoczone wymienionymi poleceniami:



Podobnie działają polecenia `\hfil` i `\hfill` wewnątrz `\hbox`-u. Np. następująca linia

```
|          [Tekst]          | powstała po wpisaniu sekwencji: \\hbox to .6\hsize{|\$|\hfil \fbox {Tekst} \hfil\hfil$|}$ powstała ...
```

Obramowanie estetycznie efektowniejsze, z cieniowaniem, można zdefiniować następująco:

```
\long\def\BaseBlock#1#2#3#4#5%
% #1 wciecie od gory, #2 grubosc z boku,
% #3 wciecie u dolu, #4 grubosc z dolu,
% #5 tekst
{\vbox{\setbox0=\hbox{#5} \offinterlineskip
\hbox{\copy0\dimen0=\ht0 \advance\dimen0 by -#1
\vrule height \dimen0 width #2}
\hbox{\hskip#3\dimen0=\wd0
\advance\dimen0 by -#3 \advance\dimen0 by #2
\vrule height #4 width \dimen0} }}
\long\def
\Shadow#1{\BaseBlock{4pt}{2pt}{4pt}{5pt}{#1}}
```

Hiszpańskie znaki j i $¿$ w T_EX-u koduje się jako `!'` i `?`. Szwedzkie $å$ należy kodować jako `\aa\AA`. Polecenie `\ss` czyni niemieckie „eszet” (β). Ponadto mamy jeszcze znaki \mathbb{C} , \mathbb{O} , \mathbb{E} , \mathbb{A} , \mathbb{I} i \mathbb{O} zapisywane poleceniami `\OE`, `\oe`, `\AE`, `\ae`, `\OI` i `\oi`.

Określona w ten sposób makrodefinicję woła się np. tak:

```
$$\Shadow{\Frame{5mm}{1pt}{\hsize=0.7\hsize
```

```
Hiszpa/nskie ... }}$$,
```

gdzie nowe `\hsize` oznacza teraz szerokość ramki. Zamiast `\Frame{}{}{}` możemy też tutaj użyć `\ramka{}{}`.

Wiele przydatnych zastosowań ma T_EX-owa komenda `\loop` o konstrukcji:

```
\loop rozkazy stałe \if ... warunkowe \repeat
```

gdzie tekst lub rozkazy *warunkowe* wykonywane są tylko wtedy, gdy warunek `...` po `\if` jest spełniony. Oto jak można wykorzystać taką pętlę do wyrysowania piramidy kul:



Uzyskuje się ją za pośrednictwem makrodefinicji:

```
\def\Triangle#1{ \def\bull{ \count1=0 \loop \edef\bull{\bullet\bullet}
\ifnum \count1<#1 \advance\count1 by 1 \centerline{\bull} \vskip-9.2pt
\repeat \vskip9.2pt\relax}}
```

i wywołania `\Triangle6`. (Dla czcionki o rozmiarze 10pt liczby 9.2 w tej definicji należy zamienić na 7.7).

Macierz z opisem wierszy i kolumn dostaje się po
`$$\bordermatrix{a&b&c\cr d&e&f\cr g&h&i\cr}$$:`

$$\begin{matrix} a & b & c \\ d & \begin{pmatrix} e & f \end{pmatrix} \\ g & \begin{pmatrix} h & i \end{pmatrix} \end{matrix}$$

Obramowanie w formie bloku możemy zdefiniować korzystając z `\loop` w celu złożenia zaczerpionego pasa z wąskich linii:

```
\long\def\Blok#1#2#3{ \fboxsep=7mm \dimen1=.4pt
% #1 ilosc linii, #2 szerokosc bloku, #3 tekst
\count1=0\hsize=#2 \setbox0=\hbox{\fbox{\vbox{#3}}}
\setbox0=\hbox{\raise\dp0\box0}
\dimen0=\ht0 \dimen2=\wd0
```

```
\loop\setbox0\hbox{\raise\dimen1\box0\rule{\dimen1}
  {\dimen0}\hskip-\dimen2\rule{\dimen2}{\dimen1}}
\advance\count1by1 \ifnum\count1<#1\repeat \box0}
```

Użycie tej makrodefinicji może wyglądać np. tak: `$$\Blok{15}{.7\hspace}{\noindent \bf Macierz z ... }$$`.

W L^AT_EX-u (polskiej wersji L^AT_EX-a) polskie znaki koduje się poprzez poprzedzenie odpowiedniej litery ASCII znakiem skośnej kreski (/), zaś samą kreskę jako //. Możemy przystosować do tego „standardu” typowy (albo i nietypowy) system T_EX przygotowując następujące definicje:

```
\def\Slash{/} \catcode'\/= \active
\def/#1{\protect\KMB#1}%       zabezp. w argumentach

\def\KMB#1{\ifmmode\Slash\ifx/#1\Slash\else#1\fi% /
\else\ifx/#1\Slash\else\leavevmode\if#1% // --> /
a{\hbox{\kern.1em\hbox{\char24} \kern-.9em#1}}% /a
\else\if#1A{\c#1}}% /A
\else\if#1e{\c#1}}\else\if#1E{{\c#1}}% /e /E
\else\if#1l{\char32l}\else\if#1L{\L}}% /l /L
\else\if#1x{\accent"5F z}}% /z
\else\if#1X{\accent"5F Z}}% /Z
\else\if#1"{\setbox0=\hbox{\lower1.3ex\hbox{"\kern%
-.04em\kern6\fontdimen1\font}}\dp0=0pt\box0}% ,,
\else\accent19#1\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi\fi
% inne znaki otrzymują akcent standardowy: \'{...}
% Rozszerzenie definicji plain.tex o \do\ /
% potrzebne dla \verb[*] i \begin/end{verbatim[*]}
\def\dospecials{\do\ \do\\\do{\do}\do{\$}\do{\&};
\do{\#\do\^\do\~\do\^K\do\_ \do\^A\do%\do\~\do\/}}
```

W definicji tej uwzględniono dodatkowo „polski” otwierający cudzysłów kodowany jako /". Rozszerzeniu wymagała również lista `\dospecials` tak, by polecenia drukowania literalnego traktowały / jako ukośną kreskę. Ponieważ / jest teraz znakiem kontrolnym (aktywnym) z argumentem, należy unikać stosowania go przed {, gdyż T_EX pozabawia wszystkie argumenty okalających je nawiasów klamrowych. I chociaż po /{...} cała zawartość ... będzie zwrócona bez uszczerbku, to jednak nie będzie ona już zawarta w klamry! /X/x używa się dla Źź. W trybie matematycznym znak / zachowuje się jak dotąd (ale uwaga na /{...}, które lepiej zapisać {\...}, /\null{...} albo \slash{...}). Gdy zechcemy umieścić go w tekście, to musimy napisać polecenie `\slash`⁷, `$/ $` albo `//` (podwojoną kreskę uzyskamy po `////`). Kodowanie korekcyjki italikowej (`\/`) pozostaje bez zmiany.

⁷`\slash` jest w L^AT_EX-u zdefiniowane jako / z możliwością złamania wiersza bezpośrednio po nim. Nie polecamy naszego `\Slash` do stosowania ze względu na jego niestandardowość

18. Jeszcze kilka użytecznych poleceń

- `\pretolerance=10000` — akceptacja paragrafów bez dzielenia na sylaby
- `\tolerance=200` — zwiększenie liczby 200 powoduje tolerowanie większych odstępów między wyrazami
- `\hyphenation{tek-s-tu jesz-cze ...}` — sposób na wskazanie miejsc podziału na sylaby (w preambule; w tekście wpisujemy indywidualnie polecenie `\-`, a nie sam dywiz, np. `tek\s\tu jesz\cze`). Jeśli nie chcemy dzielenia pomiędzy wyrazami, to zamiast odstępu piszemy tyldę (np. `1992~r.`)
- `\showhyphens{wyraz wyrazy}` — na monitorze i w zbiorze `.log` pojawi się komunikat o tym, jak \TeX dzieli podane wyrazy; w tym wypadku dostalibyśmy:
- ```
[] \tenrm wy-raz wy-razy
```
- `\show\command` — pokazuje znaczenie (definicję, albo zawartość rejestru) komendy na monitorze i w pliku `.log`; np. `\show\centerline`
- `\string\command` — np. `\tt\string\par` daje `\par`
- `\message{...}` — wysłanie komunikatu ... na terminal i do zbioru `.log`
- `\typeout{...}` — j.w.
- `\typein[\komenda]{...}` — j.w., ale  $\TeX$  czeka na odpowiedź z klawiatury. Odpowiedź ta (zakończona klawiszem `<ENTER>`) jest traktowana jak fragment zbioru źródłowego w miejscu `\typein`. Jeśli podano opcjonalny argument `\komenda`, to wpisany komunikat zostaje przypisany w całości temu argumentowi i można go wywołać tak powstałym nowym poleceniem `\komenda`. Polecenie `\typein{...}` może być wykorzystane do roboczego wpisywania rozkazu `\includeonly{...}` (s. 45)
- `\special{...}` — wstawia ... do pliku `.dvi` (mogą to być instrukcje sterujące urządzenia piszącego, np. wysłanie rysunku do drukarki laserowej)
- `\relax` — nie rób nic (kończy poszukiwanie argumentu poprzedzających poleceń)
- `\uppercase{...}`, `\lowercase{...}` — tekst ... jest wypisywany tylko dużymi lub małymi literami
- `\obeylines`, `\obeyspaces` — polecenia literalnego traktowania zmian wiersza (każdy wiersz jest traktowany tak, jak paragraf!) i odstępów w tekście źródłowym
- `\everyOBJ{...}` — ... jest wykonywane po każdym wystąpieniu `OBJ`, przy czym `OBJ` może być: `cr`, `par`, `hbox`, `vbox`, `math`, `display` i `job` (to ostatnie przeznaczone w zasadzie dla  $\text{\LaTeX}$ ). Przykład: `\everycr{\newline}`
- `\vtop{...}` — taki, jak `\vbox{...}`, ale z poziomym wyrównywaniem od góry
- `\leaders\hbox{$\star$}\hfill\null\` — `*****`  
(przestrzeń wypełniana jest `hboxem` ciasno od lewej; odmiana z centrowaniem to `\cleaders`, a z równomiernym rozkładem — `\xleaders`)
- `\phantom{...}`, `\hphantom{...}` i `\vphantom{...}` — pusty `box` o rozmiarach obiektu ..., o wysokości `0pt` i o długości `0pt` (odpowiednio)
- `\smash{...}` — oryginalna zawartość obiektu ..., ale o wysokości `0pt`
- `\adjust{\vskip3mm\centerline{Co/s}\vskip5mm}` — robi odstęp wewnątrz para-

Coś

grafu po wierszu zawierającym `\adjust` i wstawia centrowane „Coś”

`\advance`, `\multiply`, `\divide` — operacje zwiększania, mnożenia i dzielenia licznika lub wymiaru. Przykłady: `\advance\textheight by -1cm` (słowo kluczowe by można opuścić), `\divide\count0 by 6`  
`aaa\llap{X$/$}bb\rlap{$/|}$ccc` — ~~Xaabbccc~~

Paragraf ten poprzedziliśmy poleceniami `\noindent` `\downbracefill\` `\indent`. Podkreślenia typu xy tg xy tg uzyskuje się w następujący sposób: `\underbar{xy tg}` `\underline{xy tg}`, przy tym oba polecenia można używać także w trybie matematycznym.

## 19. T<sub>E</sub>X — L<sub>A</sub>T<sub>E</sub>X

W L<sub>A</sub>T<sub>E</sub>X-u brak jest następujących (spośród ponad 900) poleceń zdefiniowanych w `plain.tex`:

|                             |                 |                             |                       |                            |                       |
|-----------------------------|-----------------|-----------------------------|-----------------------|----------------------------|-----------------------|
| <code>\cleartabs</code>     | <code>\+</code> | <code>\tabs</code>          | <code>\tabdone</code> | <code>\tabset</code>       | <code>\settabs</code> |
| <code>\tabalign</code>      |                 | <code>\footnotestrut</code> |                       | <code>\endinsert</code>    |                       |
| <code>\topins</code>        |                 | <code>\topinsert</code>     |                       | <code>\midinsert</code>    |                       |
| <code>\pageno</code>        |                 | <code>\nopagenumbers</code> |                       | <code>\makeheadline</code> |                       |
| <code>\headline</code>      |                 | <code>\advancepageno</code> |                       | <code>\makefootline</code> |                       |
| <code>\footline</code>      |                 | <code>\pageinsert</code>    |                       | <code>\dosupereject</code> |                       |
| <code>\normalbottoms</code> |                 | <code>\pagecontents</code>  |                       | <code>\plainoutput</code>  |                       |
| <code>\folio</code>         |                 | <code>\pagebody</code>      |                       | <code>\vfootnote</code>    |                       |
| <code>\magnification</code> |                 | <code>\beginsection</code>  |                       | <code>\oldstyle</code>     |                       |
| <code>\eqalign</code>       |                 | <code>\eqalignno</code>     |                       | <code>\leqalignno</code>   |                       |
| <code>\fivei</code>         |                 | <code>\fivebf</code>        |                       | <code>\sevensty</code>     |                       |
| <code>\fiverm</code>        |                 | <code>\seveni</code>        |                       | <code>\sevenbf</code>      | <code>\end</code>     |
| <code>\fivesy</code>        |                 | <code>\sevenrm</code>       |                       | <code>\teni</code>         | <code>\bye</code>     |

Ponadto zmieniono istotnie m.in. polecenia `\line` i `\centering`.

**Nowy format** możemy przygotować wołając T<sub>E</sub>X-a w następujący sposób:

```
tex plain/i albo tex lplain/i
```

(zostanie wczytany zbiór `plain.tex` albo `lplain` z katalogu bieżącego albo z `\texinput` bądź `\inputs`). Plik `lplain.tex` jest modyfikacją `plain.tex`, która zawiera m.in. polecenie wczytania zbioru `latex.tex` (ten z kolei czyta jeszcze plik `lfonts.tex`). Jeżeli program wykonawczy ma inną nazwę (np. `texml.exe`), to naturalnie pierwszym słowem będzie jego nazwa, a nie `tex`. W niektórych wersjach (np. sbT<sub>E</sub>X) jest oddzielny program do tworzenia formatów: `initex.exe`, i wtedy format tworzymy pisząc po prostu np. `initex plain`. Po przetworzeniu tego zbioru, gdy pojawi się komunikat `*` można napisać `\dump`, albo poprzedzić to innymi poleceniami, np. poleceniem wczytania dodatkowego zbioru (nakładki, stylu, lokalnej modyfikacji) poleceniem `\input zbior[.ext]`. Nowy format otrzymuje rozszerzenie `.fmt` (i niekiedy jest zapisywany wprost do katalogu

`\texfmts`, albo trzeba go umieścić tam bądź w katalogu `\formats`, zależnie od konfiguracji).

Gdy mamy już zbiór `plain.fmt`, wtedy nasz własny plik `plik.tex` przetworzymywołając T<sub>E</sub>X-a następująco:

```
tex &plain plik albo tex &lplain plik
```

przy czym druga forma jest odpowiednia do pliku L<sup>A</sup>T<sub>E</sub>X-owego. Przetwarzanie z formatem jest równoważne przetwarzaniu bezformatowemu, ale ze zbiorem `plain.tex` bądź `lplain.tex` rozszerzonym o `plik.tex` (i oczywiście jest znacznie szybsze od tego drugiego).

## 20. Podział pliku źródłowego

Przygotowanie ostatecznej wersji dokumentu wymaga zwykle wielokrotnego przetwarzania T<sub>E</sub>X-em. W przypadku obszerniejszych tekstów staje się to zbyt czasochłonne — znacznie przyjemniej pracuje się z małymi blokami. Z dużego pliku możemy zawsze wydzielić wiele mniejszych, np. zgodnie z rozdziałami, a T<sub>E</sub>X-owi polecić wczytywać kolejne partie rozkazem

```
\input{plik_i} lub
\include{plik_i},
```

wstawianym w odpowiednie miejsca przetwarzanego pliku (mówmy, głównego). W powyższym, `plik_i` jest oczywiście nazwą pliku zawierającego część naszego dokumentu. Efekt jest taki, jak gdyby `plik_i` został na powrót wstawiony do głównego, z tym że po `\include` skład tekstu zacznie się od nowej strony. Jest to niekorzystne, ale za to istnieje dodatkowa możliwość umieszczenia w preambule (przed `\begin{document}`) polecenia

```
\includeonly{plik_1,...,plik_N},
```

które będzie oznaczało, że jedynie `plik_1, ..., plik_N` spośród wszystkich wołanych poleceniem `\include` będą rzeczywiście przetwarzane. Oznacza to także, że jeśli napiszemy `\includeonly{}` (z pustym argumentem), to żaden z takich plików nie zostanie wczytany, ale fragment naszego dokumentu będzie przetworzony według stanu z ostatniego pełnego przetwarzania (numeracja stron, sekcji, odwołania, przypisy itp. pamiętane są w plikach `plik_i.aux`).

## 21. Jak złożyć kilka stron na jednej kartce (w pliku .dvi)

Kiedy przygotowujemy publikację o stronach małego formatu możemy zapragnąć wydrukować roboczo po dwie strony na typową kartkę. „Ręczne” składanie dwóch wydruków na tej samej kartce jest w praktyce uciążliwe. Można jednak w tym celu zmodyfikować program L<sup>A</sup>T<sub>E</sub>X (zbiór źródłowy `latex.tex`). Nie polecamy oczywiście modyfikacji samego L<sup>A</sup>T<sub>E</sub>X-a, ponieważ mogłoby to prowadzić do nieprzyjemnych niespodzianek, ale możemy

niektóre jego funkcje przedefiniować we własnym pliku źródłowym, bądź w prywatnym pliku stylu (z rozszerzeniem `.sty`).

Oto przykład, który autor opracował dla własnych potrzeb przygotowując tę publikację. Wiersze oznaczone na poniższym wydruku `%KMB` są istotnymi zmianami w stosunku do odpowiedniego fragmentu `latex.tex`. Pierwsze i ostatnie polecenie dotyczy zmiany kategorii znaku `@` na literę i powrót do poprzedniego znaczenia. Rejestr `\LR` zawiera informację, która strona ma być składana na kartkę (`L` — lewa, `R` — prawa), zaś gdy wpisujemy tam cokolwiek innego (domyślnie jest `0`), to wydruk jest robiony po jednej stronie na kartkę (tak, jak bez modyfikacji). Jeśli więc w naszym pliku źródłowym umieścimy w pewnym miejscu polecenie `\let\LR=L`, to odtąd w pliku `.dvi` będą zapisane po dwie strony na kartkę. Możemy też w dalszej części napisać `\let\LR=0`, co spowoduje przejście do trybu poprzedniego.

```

\makeatletter %zb/edne gdy w pliku .STY %KMB
\let\LR=0 \newbox\@leftpage %KMB
\def\@fullline{\hbox to 2.4\hsize} %KMB

\def\@outputpage{\begingroup\catcode'\ =10
 \let-\@dischyp \let'\@acci
 \let'\@accii \let=\@acciii
 \if@specialpage\global\@specialpagefalse
 \nameuse{ps@\@specialstyle}\fi
 \if@twoside
 \ifodd\count\z@ \let\@thehead\@oddhead
 \let\@thefoot\@oddfont
 \let\@themargin\oddsidemargin
 \else \let\@thehead\@evenhead
 \let\@thefoot\@evenfont
 \let\@themargin\evensidemargin
 \fi\fi
% \shipout %KMB
\gdef\@pagebox{ %KMB
\ vbox{\normalsize \baselineskip\z@ \lineskip\z@
 \let\par\@par
 \vskip \topmargin \moveright\@themargin
 \vbox{\setbox\@tempboxa
 \vbox to\headheight{\vfil \hbox to\textwidth
 {\let\label\@gobble \let\index\@gobble
 \@thehead}}}
 \dp\@tempboxa\z@
 \box\@tempboxa
 \vskip \headsep
 \box\@outputbox
 \baselineskip\footskip
 \hbox to\textwidth{\let\label\@gobble
 \let\index\@gobble \@thefoot}}}
```



```

} %KMB
\if L\LR \global\setbox\@leftpage=\@pagebox %KMB
 \global\let\LR=R %KMB
\else\if R\LR \global\let\LR=L \shipout %KMB
 \@fullline{\box\@leftpage\hss\@pagebox}%KMB
 \else \shipout\@pagebox\fi\fi %KMB

\global\@colht\textheight\endgroup
\stepcounter{page}\let\firstmark\botmark}
\makeatother %usun/a/c je/sli w pliku .STY %KMB

```

Działanie tej procedury polega na zapamiętaniu pierwszej strony z dwóch kolejnych (przy `\LR=L`) w *boxie* `\@leftpage` do czasu złożenia następnej. Gdy i druga jest już gotowa, to obie są umieszczane obok siebie w *hboxie* `\@fullline` i dopiero teraz wysyłane do pliku `.dvi` poleceniem `\shipout`. Szerokość `\@fullline` ustalono tutaj arbitralnie na `2.4 \hspace`, ale można ją zmienić w zależności od potrzeb (przede wszystkim zależnie od szerokości pojedynczych stron). Czytelnik łatwo domyśli się, jak można złożyć trzy, albo więcej stron obok siebie, albo jedną nad drugą (trzeba je zapamiętać, a potem odpowiadające im *boxy* poustawiać w końcowym *hboxie* lub *vboxie*).

## 22. Jeszcze o fontach

### 22.1 Rodziny

System T<sub>E</sub>X ma rejestr *rodzin* fontów (maksymalnie 16, o numerach `\fam` od 0 do 15). W L<sup>A</sup>T<sub>E</sub>X-u zdefiniowano (w zbiorze `lfonts.tex`) następujące rodziny:

|                                            |                                               |
|--------------------------------------------|-----------------------------------------------|
| 0 <code>\rm</code> ( <code>\rmfam</code> ) | 6 <code>\bf</code> ( <code>\bffam</code> )    |
| 1 <code>\mit</code>                        | 7 <code>\tt</code> ( <code>\ttfam</code> )    |
| 2 <code>\cal</code> (symbole <i>math</i> ) | 8 <code>\sf</code> ( <code>\sffam</code> )    |
| 3 <code>ex</code> rozszerz. <i>math</i>    | 9 <code>\sc</code> ( <code>\scfam</code> )    |
| 4 <code>\it</code> ( <code>\itfam</code> ) | 10 <code>\ly</code> ( <code>\lyfam</code> )   |
| 5 <code>\sl</code> ( <code>\slfam</code> ) | 11 <code>\uit</code> ( <code>\uitfam</code> ) |

Rodzina 11 (wyprostowana itałika, ang. upright albo unslanted italic) występuje dopiero w późniejszych wersjach L<sup>A</sup>T<sub>E</sub>X-a.

Nową rodzinę (o kolejnym wolnym numerze) możemy założyć pisząc `\newfam \nazwarodziny`. Polecenie `\the \fam` poda nam numer aktualnie używanej rodziny. Np. `{\bf \the \fam}` daje **6**. Numer ten zmienia się po poleceniach `\it`, `\sc` itp.

W każdej rodzinie może być określony `\textfont` — do składania zwykłych tekstów (np. `\textfont0=\tenrm`, `\textfont1=\tenmi`), `\scriptfont` — do pisania wskaźników i potęg i `\scriptscriptfont` — do składu wykładników i wskaźników drugiego rzędu.

### 22.2 Parametry fontów

Każdy font, prócz kodu graficznego znaków (zwykle do 128), zawiera wiele parametrów ogólnych, ważnych przy składaniu tekstu (`tex.exe` nie korzysta w ogóle z kodów

Typowy zestaw fontów T<sub>E</sub>X-a

| Font ↓ $m \Rightarrow$ | 0 | 1/2 | 1 | 2 | 3 | 4 | 5 | Font ↓ $m \Rightarrow$ | 0 | 1/2 | 1 | 2 | 3 |
|------------------------|---|-----|---|---|---|---|---|------------------------|---|-----|---|---|---|
| CIRCLE10               | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSS8                  | ✓ | ✓   | ✓ |   |   |
| CIRCLEW1               | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSS9                  | ✓ | ✓   | ✓ |   |   |
| CMB10                  | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSSBX10               | ✓ | ✓   | ✓ |   |   |
| CMBSY10                | ✓ | ✓   | ✓ |   |   |   |   | CMSSDC10               | ✓ | ✓   | ✓ |   |   |
| CMBX10                 | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSSI10                | ✓ | ✓   | ✓ |   |   |
| CMBX12                 | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSSI12                | ✓ | ✓   | ✓ |   |   |
| CMBX5                  | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSSI17                | ✓ | ✓   | ✓ |   |   |
| CMBX6                  | ✓ | ✓   | ✓ |   |   |   |   | CMSSI8                 | ✓ | ✓   | ✓ |   |   |
| CMBX7                  | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSSI9                 | ✓ | ✓   | ✓ |   |   |
| CMBX8                  | ✓ | ✓   | ✓ |   |   |   |   | CMSSQ8                 | ✓ | ✓   | ✓ |   |   |
| CMBX9                  | ✓ | ✓   | ✓ |   |   |   |   | CMSSQ18                | ✓ | ✓   | ✓ |   |   |
| CMBXSL10               | ✓ | ✓   | ✓ |   |   |   |   | CMSY10                 | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMBXT10                | ✓ | ✓   | ✓ |   |   |   |   | CMSY5                  | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMCSC10                | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSY6                  | ✓ | ✓   | ✓ |   |   |
| CMDUNH10               | ✓ | ✓   | ✓ |   |   |   |   | CMSY7                  | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMEX10                 | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMSY8                  | ✓ | ✓   | ✓ |   |   |
| CMFF10                 | ✓ | ✓   | ✓ |   |   |   |   | CMSY9                  | ✓ | ✓   | ✓ |   |   |
| CMFI10                 | ✓ | ✓   | ✓ |   |   |   |   | CMTCSC10               | ✓ | ✓   | ✓ |   |   |
| CMFIB8                 | ✓ | ✓   | ✓ |   |   |   |   | CMTEX10                | ✓ | ✓   | ✓ |   |   |
| CMITT10                | ✓ | ✓   | ✓ |   |   |   |   | CMTEX8                 | ✓ | ✓   | ✓ |   |   |
| CMMI10                 | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMTEX9                 | ✓ | ✓   | ✓ |   |   |
| CMMI12                 | ✓ | ✓   | ✓ |   |   |   |   | CMTI10                 | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMMI5                  | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMTI12                 | ✓ | ✓   | ✓ |   |   |
| CMMI6                  | ✓ | ✓   | ✓ |   |   |   |   | CMTI7                  | ✓ | ✓   | ✓ |   |   |
| CMMI7                  | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMTI8                  | ✓ | ✓   | ✓ |   |   |
| CMMI8                  | ✓ | ✓   | ✓ |   |   |   |   | CMTI9                  | ✓ | ✓   | ✓ |   |   |
| CMMI9                  | ✓ | ✓   | ✓ |   |   |   |   | CMTT10                 | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMMIB10                | ✓ | ✓   | ✓ |   |   |   |   | CMTT12                 | ✓ | ✓   | ✓ |   |   |
| CMR10                  | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMTT8                  | ✓ | ✓   | ✓ |   |   |
| CMR12                  | ✓ | ✓   | ✓ |   |   |   |   | CMTT9                  | ✓ | ✓   | ✓ |   |   |
| CMR17                  | ✓ | ✓   | ✓ |   |   |   |   | CMU10                  | ✓ | ✓   | ✓ |   |   |
| CMR5                   | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | CMVTT10                | ✓ | ✓   | ✓ |   |   |
| CMR6                   | ✓ | ✓   | ✓ |   |   |   |   | LASY10                 | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMR7                   | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | LASY5                  | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMR8                   | ✓ | ✓   | ✓ |   |   |   |   | LASY6                  | ✓ | ✓   | ✓ |   |   |
| CMR9                   | ✓ | ✓   | ✓ |   |   |   |   | LASY7                  | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMSL10                 | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | LASY8                  | ✓ | ✓   | ✓ |   |   |
| CMSL12                 | ✓ | ✓   | ✓ |   |   |   |   | LASY9                  | ✓ | ✓   | ✓ |   |   |
| CMSL8                  | ✓ | ✓   | ✓ |   |   |   |   | LASYB10                | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMSL9                  | ✓ | ✓   | ✓ |   |   |   |   | LINE10                 | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMSLTT10               | ✓ | ✓   | ✓ |   |   |   |   | LINEW10                | ✓ | ✓   | ✓ | ✓ | ✓ |
| CMSS10                 | ✓ | ✓   | ✓ | ✓ | ✓ | ✓ | ✓ | LOGO10                 | ✓ | ✓   | ✓ |   |   |
| CMSS12                 | ✓ | ✓   | ✓ |   |   |   |   | CMINCH                 | ✓ |     |   |   |   |
| CMSS17                 | ✓ | ✓   | ✓ |   |   |   |   | MANUAL                 | ✓ |     |   |   |   |

graficznych umieszczonych w plikach `.pk`, lecz tylko z metryki fontów — pliki `.tfm`). Podstawowe parametry fontu można odczytać, albo zmienić, poleceniami `\fontdimenN` `\fontname`. Np., na skutek polecenia `\the \fontdimen2 \tenrm` otrzymujemy 3.33333pt, co jest nominalną wartością odstepu międzywyrazowego (tzw. `\spaceskip`) przy używaniu fontu `cmr10`, zaś `{\it\the\fontdimen1\font}` daje nachylenie w punktach na 1pt (*slant*, ważny przy składaniu akcentów) aktualnie używanego fontu (tu kursywa): *0.25pt*. Pierwsze siedem parametrów to: (N = 1) nachylenie, (2) `\spaceskip`, (3) dopuszczalne zwiększenie `\spaceskip`, (4) dopuszczalne zmniejszenie `\spaceskip`, (5) wysokość litery x (jednostka `ex`), (6) jednostka `em` (też `\quad`), (7) dodatkowe powiększenie `\spaceskip` dla znaków przestankowych, czyli tzw. `\xspaceskip`. Poniższa tabela pokazuje przykładowe wartości (wszystkie w jednostkach pt) wymienionych parametrów dla czcionki o rozmiarze 10 pt.

| <code>(\fontdimen N)</code> | <code>N ⇒</code>                  | 1            | 2                      | 3    | 4    | 5    | 6     | 7           |
|-----------------------------|-----------------------------------|--------------|------------------------|------|------|------|-------|-------------|
| <code>\rm</code>            | <code>(\tenrm=cmr10)</code>       | 0            | 3.33                   | 1.67 | 1.11 | 4.31 | 10    | 1.11        |
| <code>\bf</code>            | <code>(\tenbf=cmbx10)</code>      | 0            | 3.83                   | 1.92 | 1.28 | 4.44 | 11.50 | 1.28        |
| <code>\it</code>            | <code>(\tenit=cmti10)</code>      | 0.25         | 3.58                   | 1.53 | 1.02 | 4.31 | 10.22 | 1.02        |
| <code>\sl</code>            | <code>(\tensl=cmsl10)</code>      | 0.16         | 3.33                   | 1.67 | 1.11 | 6.31 | 10    | 1.11        |
| <code>\tt</code>            | <code>(\tentt=cmtt10)</code>      | 0            | 5.25                   | 0    | 0    | 4.31 | 10.49 | 5.25        |
| <code>\mit</code>           | <code>(\tenmi=cmmi10)</code>      | 0.25         | 0                      | 0    | 0    | 4.31 | 10    | 0           |
| <code>\cal</code>           | <code>(\textfont2, cmsy10)</code> | 0.25         | 0                      | 0    | 0    | 4.31 | 10    | 0           |
| <code>ex</code>             | <code>(\textfont3, cmex10)</code> | 0            | 0                      | 0    | 0    | 4.32 | 10    | 0           |
|                             |                                   | <i>slant</i> | <i>(spaceskip + -)</i> |      |      | 1ex  | 1em   | <i>(x+)</i> |

Z przytoczonej tabelki wynika, że dla `\rm` (przy 10pt) `\spaceskip=3.33pt plus 1.67pt minus 1.11pt`. Możemy jednak napisać takie polecenie z innymi parametrami. Podobnie, pisząc `\xspaceskip=3.33pt plus2.22pt` podwajamy standardową tolerancję zawartą w `\fontdimen7`.

## 22.3 Bezpośredni dostęp do fontów

W  $\text{\LaTeX}$ -u, prócz rodzin typu `\rm`, `\it` ... poszczególne kroje i rozmiary są dostępne bezpośrednio w stosunkowo szerokim repertuarze i tylko w nielicznych przypadkach trzeba załadować dodatkową czcionkę ( $\text{\TeX}$ -owym poleceniem `\font\nazwa= ... [scaled \magstep m]`).

Niektóre kompozycje kroju i rozmiaru doładowywane automatycznie nie będą akceptowane w trybie *math*. Remedium na ten niedostatek może być polecenie typu

```
\load{\footnotesize}{\sf}
```

Nazwy fontów dostępnych w różnych rozmiarach podajemy na następnej stronie. W tabelkach tych `D` oznacza, że dany font jest dostępny po doładowaniu (które odbywa się automatycznie), ale nie przez własną nazwę a tylko w rodzinie. Możemy np. bez obawy pisać `\sc`, ale nie `\tensc`. W rodzinie *lasy* (`\ly`) występują specjalne symbole  $\text{\LaTeX}$ -a. W ostatniej kolumnie podaliśmy przyporządkowanie fontów rozmiarom standardowym (`\tiny` i in.) dla przypadku, gdy w `\documentstyle` nie specyfikowano opcji rozmiaru (11pt ani 12pt), czyli efektywnie przy rozmiarze domyślnym 10pt. Poza wymienionymi w tabelce w serii o rozmiarze 10 pt załadowano także rozszerzenie

*math* `\tenex=cmex10` oraz fonty linii i kół: `\tenln = line10`, `\tenlnw = linew10`, `\tencirc = circle10`, `\tencircw = circlew10`. Czcionki te wczytano poleceniami typu `\font\tenex = cmex10` podczas inicjacji L<sup>A</sup>T<sub>E</sub>X-a i każda może być wywołana w tekście źródłowym przez przypisanie w ten sposób nazwy. **Np., polecenie `\frtbnf` Np., polecenie ...} daje efekt jak w tym zdaniu.** Inne kroje i rozmiary czcionek, w szczególności nowo utworzone METAFONT-em, muszą być doładowane.

|      |                      |                      |                      |                      |                      |                            |
|------|----------------------|----------------------|----------------------|----------------------|----------------------|----------------------------|
|      | <code>\rm</code>     | <code>\mit</code>    | <code>\cal</code>    | <code>\it</code>     | <code>\sl</code>     |                            |
| 5 pt | <code>\fivrm</code>  | <code>\fivmi</code>  | <code>\fivsy</code>  |                      |                      | <code>\vpt</code>          |
| 6 pt | <code>\sixrm</code>  | <code>\sixmi</code>  | <code>\sixsy</code>  |                      |                      | <code>\vipt</code>         |
| 7 pt | <code>\sevrm</code>  | <code>\sevmi</code>  | <code>\sevsy</code>  | D                    | D                    | <code>\viipt</code>        |
| 8 pt | <code>\egtrm</code>  | <code>\egtmi</code>  | <code>\egtsy</code>  | <code>\egtit</code>  | D                    | <code>\viipt</code>        |
| 9 pt | <code>\ninrm</code>  | <code>\ninmi</code>  | <code>\ninsy</code>  | <code>\ninit</code>  | D                    | <code>\ixpt</code>         |
| 10pt | <code>\tenrm</code>  | <code>\tenmi</code>  | <code>\tensy</code>  | <code>\tenit</code>  | <code>\tensl</code>  | <code>\xpt</code>          |
| 11pt | <code>\elevrm</code> | <code>\elevmi</code> | <code>\elevsy</code> | <code>\elevit</code> | <code>\elevsl</code> | <code>\xipt</code>         |
| 12pt | <code>\twlrm</code>  | <code>\twlmi</code>  | <code>\twlsy</code>  | <code>\twlit</code>  | <code>\twlsl</code>  | <code>\xiipt</code>        |
| 14pt | <code>\frtrm</code>  | <code>\frtmi</code>  | <code>\frtsy</code>  | D                    | D                    | <code>\xivpt</code>        |
| 17pt | <code>\svtrm</code>  | <code>\svtmi</code>  | <code>\svtsy</code>  | D                    | D                    | <code>\xvipt</code>        |
| 20pt | <code>\twtyrm</code> | <code>\twtymi</code> | <code>\twtysy</code> | D                    | D                    | <code>\xxpt</code>         |
| 25pt | <code>\twfvr</code>  |                      |                      |                      |                      | <code>\xxvpt</code>        |
|      | <code>\bf</code>     | <code>\tt</code>     | <code>\sf</code>     | <code>\sc</code>     | <code>\ly</code>     | (10pt)                     |
| 5 pt | D                    |                      |                      |                      | <code>\fivly</code>  | <code>\tiny</code>         |
| 6 pt | D                    |                      |                      |                      | <code>\sixly</code>  |                            |
| 7 pt | D                    |                      |                      |                      | <code>\sevly</code>  | <code>\scriptsize</code>   |
| 8 pt | D                    | D                    | D                    | D                    | <code>\egtly</code>  | <code>\footnotesize</code> |
| 9 pt | <code>\ninbf</code>  | <code>\nintt</code>  | D                    | D                    | <code>\ninly</code>  | <code>\small</code>        |
| 10pt | <code>\tenbf</code>  | <code>\tentt</code>  | <code>\tensf</code>  | D                    | <code>\tenly</code>  | <code>\normalsize</code>   |
| 11pt | <code>\elevbf</code> | <code>\elevtt</code> | <code>\elevsf</code> | D                    | <code>\elevly</code> |                            |
| 12pt | <code>\twlbf</code>  | <code>\twlftt</code> | <code>\twlfsf</code> | D                    | <code>\twlly</code>  | <code>\large</code>        |
| 14pt | <code>\frtnbf</code> | D                    | D                    | D                    | <code>\frtnly</code> | <code>\Large</code>        |
| 17pt | <code>\svtnbf</code> | D                    | D                    | D                    | <code>\svtnly</code> | <code>\LARGE</code>        |
| 20pt | D                    | D                    | D                    | D                    | <code>\twtyly</code> | <code>\huge</code>         |
| 25pt | D                    |                      |                      |                      |                      | <code>\Huge</code>         |

Której czcionki T<sub>E</sub>X aktualnie używa możemy dowiedzieć się z `\fontnamefont`. Np., po `\fontname\twtysy; {\twfvr\fontname\font.}` otrzymujemy: `cmsy10 at 20.73999pt; cmr10 at 24.87999pt.`

Poniżej przedstawiamy tabele znaków w ważniejszych fontach L<sup>A</sup>T<sub>E</sub>X-a. Podobne tabele dla antyki i symboli matematycznych znajdują się w p. 6.

Komplet znaków wyprostowanej kursywy (cmu10)

|     | 0 | 1  | 2  | 3  | 4   | 5   | 6 | 7 | 8 | 9 |
|-----|---|----|----|----|-----|-----|---|---|---|---|
| x   | Γ | Δ  | Θ  | Λ  | Ξ   | Π   | Σ | Υ | Φ | Ψ |
| 1x  | Ω | ff | fi | fl | ffi | ffl | ι | ι | ϵ | ϵ |
| 2x  | ˘ | ˘  | ˘  | ˘  | ˘   | ˘   | ˘ | ˘ | ˘ | ˘ |
| 3x  | œ | ø  | -  | !  | ”   | #   | £ | % | € | ’ |
| 4x  | ( | )  | *  | +  | ,   | -   | . | / | 0 | 1 |
| 5x  | 2 | 3  | 4  | 5  | 6   | 7   | 8 | 9 | : | ; |
| 6x  | i | =  | ¿  | ?  | @   | A   | B | C | D | E |
| 7x  | F | G  | H  | I  | J   | K   | L | M | N | O |
| 8x  | P | Q  | R  | S  | T   | U   | V | W | X | Y |
| 9x  | Z | [  | “  | ]  | ^   | .   | ’ | a | b | c |
| 10x | d | e  | f  | g  | h   | i   | j | k | l | m |
| 11x | n | o  | p  | q  | r   | s   | t | u | v | w |
| 12x | x | y  | z  | -  | —   | ”   | ˘ | ˘ | ˘ | ˘ |

Symbole L<sup>A</sup>T<sub>E</sub>X-a (lasy10)

|    | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----|---|---|---|---|---|---|---|---|---|---|
| x  |   | ◁ | ◁ | ▷ | ▷ |   |   |   | ∅ | ∞ |
| 4x | < | > | ^ | v |   |   |   |   | ∅ | ∞ |
| 5x | □ | ◇ |   |   |   |   |   |   | ~ | ~ |
| 6x | □ | □ |   |   |   |   |   |   |   |   |

Komplet znaków w foncie line10 (L<sup>A</sup>T<sub>E</sub>X)

|     | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|-----|---|---|---|---|---|---|---|---|---|---|
| x   | / | / | / | / | / | / |   |   | / | ↗ |
| 1x  | / | ↗ | / | ↘ | ↘ | ↘ | — | / | ↘ | / |
| 2x  | / | ↘ |   | ↗ | — | ↗ | — | ↘ | / | ↘ |
| 3x  | ↘ |   | — | — | — | — |   | — |   | ↘ |
| 4x  | — | ↗ | ↘ | ↗ | — | ↘ |   | ↗ |   | ↘ |
| 5x  |   | ↘ | ↘ | ↘ | ↘ | ↘ |   | ↘ | ↘ | ↘ |
| 6x  |   | ↗ | ↘ | ↘ | ↘ | ↘ | — | ↘ | ↘ | ↘ |
| 7x  |   |   | — | ↘ | ↘ | ↘ | — | ↘ | ↘ | ↘ |
| 8x  | — | — | ↗ | ↘ | ↘ | ↘ |   | ↘ | — | ↘ |
| 9x  | — |   | ↘ | ↘ | ↘ | ↘ | — | — | — | — |
| 10x |   | ↘ |   |   | — | ↘ | ↗ | ↘ | — |   |
| 11x |   | ↘ |   | ↗ |   | ↘ |   |   |   | ↘ |
| 12x |   | ↘ | ↗ |   |   | ↘ | ↗ |   |   | ↘ |

Komplet znaków italiki *math* (*cmmi10*)

|     | 0            | 1                       | 2             | 3                        | 4             | 5           | 6                | 7               | 8           | 9          |
|-----|--------------|-------------------------|---------------|--------------------------|---------------|-------------|------------------|-----------------|-------------|------------|
| x   | $\Gamma$     | $\Delta$                | $\Theta$      | $\Lambda$                | $\Xi$         | $\Pi$       | $\Sigma$         | $\Upsilon$      | $\Phi$      | $\Psi$     |
| 1x  | $\Omega$     | $\alpha$                | $\beta$       | $\gamma$                 | $\delta$      | $\epsilon$  | $\zeta$          | $\eta$          | $\theta$    | $\iota$    |
| 2x  | $\kappa$     | $\lambda$               | $\mu$         | $\nu$                    | $\xi$         | $\pi$       | $\rho$           | $\sigma$        | $\tau$      | $\upsilon$ |
| 3x  | $\phi$       | $\chi$                  | $\psi$        | $\omega$                 | $\varepsilon$ | $\vartheta$ | $\varpi$         | $\varrho$       | $\varsigma$ | $\varphi$  |
| 4x  | $\leftarrow$ | $\overleftarrow{\quad}$ | $\rightarrow$ | $\overrightarrow{\quad}$ | $\cdot$       | $\circ$     | $\triangleright$ | $\triangleleft$ | $\circ$     | $\iota$    |
| 5x  | 2            | 3                       | 4             | 5                        | 6             | 7           | 8                | 9               | .           | ,          |
| 6x  | <            | /                       | >             | *                        | $\partial$    | A           | B                | C               | D           | E          |
| 7x  | F            | G                       | H             | I                        | J             | K           | L                | M               | N           | O          |
| 8x  | P            | Q                       | R             | S                        | T             | U           | V                | W               | X           | Y          |
| 9x  | Z            | b                       | ‡             | #                        | )             | (           | l                | a               | b           | c          |
| 10x | d            | e                       | f             | g                        | h             | i           | j                | k               | l           | m          |
| 11x | n            | o                       | p             | q                        | r             | s           | t                | u               | v           | w          |
| 12x | x            | y                       | z             | i                        | j             | ø           | $\bar{\quad}$    | $\hat{\quad}$   |             |            |

Komplet znaków rozszerzenia *math* (*cmex10*)

|     | 0        | 1        | 2       | 3        | 4        | 5        | 6        | 7        | 8               | 9               |
|-----|----------|----------|---------|----------|----------|----------|----------|----------|-----------------|-----------------|
| x   | (        | )        | [       | ]        | [        | ]        | [        | ]        | {               | }               |
| 1x  | <        | >        |         |          | /        | \        | (        | )        | (               | )               |
| 2x  | [        | ]        | [       | ]        | [        | ]        | {        | }        | <               | >               |
| 3x  | /        | \        | (       | )        | [        | ]        | [        | ]        | [               | ]               |
| 4x  | {        | }        | <       | >        | /        | \        | /        | \        | (               | )               |
| 5x  | [        | ]        | [       | ]        |          |          | r        | r        | u               | u               |
| 6x  | {        | }        | '       | '        | \        | )        | '        | '        | <               | >               |
| 7x  | $\sqcup$ | $\sqcup$ | $\oint$ | $\oint$  | $\odot$  | $\odot$  | $\oplus$ | $\oplus$ | $\otimes$       | $\otimes$       |
| 8x  | $\Sigma$ | $\Pi$    | $\int$  | $\cup$   | $\cap$   | $\oplus$ | $\wedge$ | $\vee$   | $\Sigma$        | $\Pi$           |
| 9x  | $\int$   | $\cup$   | $\cap$  | $\oplus$ | $\wedge$ | $\vee$   | $\Pi$    | $\Pi$    | $\tilde{\quad}$ | $\tilde{\quad}$ |
| 10x | $\int$   | $\sim$   | $\sim$  | $\sim$   | $\wedge$ | $\vee$   | $\Pi$    | $\Pi$    | $\tilde{\quad}$ | $\tilde{\quad}$ |
| 11x | {        | }        | ✓       | ✓        | ✓        | ✓        | ✓        |          |                 |                 |
| 12x | ↑        | ↓        | ↗       | ↘        | ↖        | ↙        | ↗        | ↘        |                 |                 |

## Komplet znaków pisma blokowego (cmss10)

|     | 0 | 1  | 2  | 3  | 4   | 5   | 6  | 7  | 8 | 9 |
|-----|---|----|----|----|-----|-----|----|----|---|---|
| x   | Γ | Δ  | Θ  | Λ  | Ξ   | Π   | Σ  | Υ  | Φ | Ψ |
| 1x  | Ω | ff | fi | fl | ffi | ffl | ι  | ϰ  | ϕ | Æ |
| 2x  | ˘ | ˘  | -  | °  | ˘   | β   | æ  | œ  | ø | ’ |
| 3x  | Œ | Ø  | -  | !  | ˘   | #   | \$ | %  | & | 1 |
| 4x  | ( | )  | *  | +  | ,   | -   | .  | /  | 0 | ; |
| 5x  | 2 | 3  | 4  | 5  | 6   | 7   | 8  | 9  | : | ; |
| 6x  | i | =  | ı  | ?  | @   | A   | B  | C  | D | E |
| 7x  | F | G  | H  | I  | J   | K   | L  | M  | N | O |
| 8x  | P | Q  | R  | S  | T   | U   | V  | W  | X | Y |
| 9x  | Z | [  | “  | ]  | ^   | ·   | ’  | a  | b | c |
| 10x | d | e  | f  | g  | h   | i   | j  | k  | l | m |
| 11x | n | o  | p  | q  | r   | s   | t  | u  | v | w |
| 12x | x | y  | z  | -  | —   | ”   | ~  | .. |   |   |

## Komplet znaków pisma maszynowego (cmtt10)

|     | 0 | 1 | 2 | 3 | 4 | 5 | 6  | 7  | 8 | 9 |
|-----|---|---|---|---|---|---|----|----|---|---|
| x   | Γ | Δ | Θ | Λ | Ξ | Π | Σ  | Υ  | Φ | Ψ |
| 1x  | Ω | ↑ | ↓ | ’ | ı | ı | ı  | ϰ  | ϕ | Æ |
| 2x  | ˘ | ˘ | - | · | ˘ | β | æ  | œ  | ø | ’ |
| 3x  | Œ | Ø | □ | ! | ˘ | # | \$ | %  | & | 1 |
| 4x  | ( | ) | * | + | , | - | .  | /  | 0 | ; |
| 5x  | 2 | 3 | 4 | 5 | 6 | 7 | 8  | 9  | : | ; |
| 6x  | < | = | > | ? | @ | A | B  | C  | D | E |
| 7x  | F | G | H | I | J | K | L  | M  | N | O |
| 8x  | P | Q | R | S | T | U | V  | W  | X | Y |
| 9x  | Z | [ | \ | ] | ^ | _ | ’  | a  | b | c |
| 10x | d | e | f | g | h | i | j  | k  | l | m |
| 11x | n | o | p | q | r | s | t  | u  | v | w |
| 12x | x | y | z | { |   | } | ~  | .. |   |   |

Komplet znaków w kursywie (*cmti10*)

|     | 0 | 1  | 2  | 3  | 4   | 5   | 6 | 7  | 8 | 9 |
|-----|---|----|----|----|-----|-----|---|----|---|---|
| x   | Γ | Δ  | Θ  | Λ  | Ξ   | Π   | Σ | Υ  | Φ | Ψ |
| 1x  | Ω | ff | fí | fl | ffi | ffl | ι | Ͽ  | ϕ | Æ |
| 2x  | ˘ | ˘  | ˘  | ˘  | ˘   | ˘   | ˘ | ˘  | ˘ | ˘ |
| 3x  | Œ | Ø  | -  | !  | ”   | #   | £ | %  | © | , |
| 4x  | ( | )  | *  | +  | ,   | -   | . | /  | 0 | 1 |
| 5x  | 2 | 3  | 4  | 5  | 6   | 7   | 8 | 9  | : | ; |
| 6x  | i | =  | ¿  | ?  | @   | A   | B | C  | D | E |
| 7x  | F | G  | H  | I  | J   | K   | L | M  | N | O |
| 8x  | P | Q  | R  | S  | T   | U   | V | W  | X | Y |
| 9x  | Z | [  | “  | ]  | ^   | ·   | ‘ | a  | b | c |
| 10x | d | e  | f  | g  | h   | i   | j | k  | l | m |
| 11x | n | o  | p  | q  | r   | s   | t | u  | v | w |
| 12x | x | y  | z  | -  | —   | ”   | ~ | .. |   |   |

## KOMPLET ZNAKÓW PISMA KAPITALIKOWEGO (CMCSC10)

|     | 0 | 1 | 2 | 3 | 4 | 5 | 6  | 7  | 8 | 9 |
|-----|---|---|---|---|---|---|----|----|---|---|
| x   | Γ | Δ | Θ | Λ | Ξ | Π | Σ  | Υ  | Φ | Ψ |
| 1x  | Ω | ↑ | ↓ | ! | ı | ı | I  | J  | ˘ | ˘ |
| 2x  | ˘ | ˘ | ˘ | ˘ | ˘ | ˘ | ˘  | ˘  | ˘ | ˘ |
| 3x  | Œ | Ø | - | ! | ” | # | \$ | %  | & | , |
| 4x  | ( | ) | * | + | , | - | .  | /  | 0 | 1 |
| 5x  | 2 | 3 | 4 | 5 | 6 | 7 | 8  | 9  | : | ; |
| 6x  | < | = | > | ? | @ | A | B  | C  | D | E |
| 7x  | F | G | H | I | J | K | L  | M  | N | O |
| 8x  | P | Q | R | S | T | U | V  | W  | X | Y |
| 9x  | Z | [ | “ | ] | ^ | · | ‘  | A  | B | C |
| 10x | D | E | F | G | H | I | J  | K  | L | M |
| 11x | N | O | P | Q | R | S | T  | U  | V | W |
| 12x | X | Y | Z | - | — | ” | ~  | .. |   |   |



## 23. Źródła

Jak już wspominaliśmy we wstępie, system  $\text{T}_{\text{E}}\text{X}$  jest „własnością publiczną”. Gdyby miał być sprzedawany w zwykłym trybie kosztowałby niewątpliwie tysiące dolarów. Prawa autorskie do programu  $\text{T}_{\text{E}}\text{X}$  należą do D.E. Knutha, ale w różnych wersjach swój udział mają różni autorzy. Np.,  $\text{s}\text{b}\text{T}_{\text{E}}\text{X}$  został opracowany przez W.G. Sullivana (E-mail: WSULIVAN@IRLEARN) i P. Breitenlohnera,  $\text{e}\text{m}\text{T}_{\text{E}}\text{X}$  — przez E. Mattesa (MATTES@AZU.INFORMATIK.UNI-STUTTGART.DE),  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  — przez L. Lamport (LAMPOR@SRC.DEC.COM).

Te implementacje, które należą do typu „public domain” (np.  $\text{s}\text{b}\text{T}_{\text{E}}\text{X}$ ,  $\text{e}\text{m}\text{T}_{\text{E}}\text{X}$ , oba z  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X-em}$ ) można kopiować, ale tylko na własne ryzyko i w zasadzie w całości. Istotne modyfikacje prowadzące do nowych wersji, które chciałoby się rozprowadzać za niewielkim wynagrodzeniem, należy bezwzględnie skonsultować z autorami. Ktokolwiek żąda zapłaty za udostępnienie powinien jasno poinformować, że jest to opłata li tylko za rozpowszechnianie, a w żadnym wypadku za programy.

Dla zainteresowanych udostępnia się także oryginalny  $\text{TEX.WEB}$  (tak w postaci książkowej, jak i drogą elektroniczną). Podobnie, dostępne są darmowe wersje  $\text{METAFont}$ -a (np.  $\text{s}\text{b}\text{MF}$ ). Większe biblioteki zbiorów dotyczących  $\text{T}_{\text{E}}\text{X}$ -a przechowuje się w kilku miejscach w USA, w Aston (Wielka Brytania) i w Heidelbergu (RFN). Więcej szczegółów — zwłaszcza o krajowych możliwościach — Czytelnik może spodziewać się zasięgając informacji u GUST-a (adres podaliśmy na s. 4).

Niektóre implementacje  $\text{T}_{\text{E}}\text{X}$ -a są wszakże licencjonowane. Znak handlowy  $\text{T}_{\text{E}}\text{X}$  należy do American Mathematical Society, a  $\text{PC T}_{\text{E}}\text{X}$  — do Personal  $\text{T}_{\text{E}}\text{X}$ , Inc.

Podczas przygotowywania tej publikacji korzystaliśmy z następujących prac:

- M. Doob**, 1992, *A Gentle introduction to  $\text{T}_{\text{E}}\text{X}$ . A Manual for Self-study*, Univ. Manitoba, Winnipeg (Canada) {rok wydania dotyczy pliku źródłowego} [E-mail: MDOOB@UOFMCC]; tłum. polskie: *Łagodne wprowadzenie do  $\text{T}_{\text{E}}\text{X}$ -a* (przekł. S. Wawrykiewicz), Polski  $\text{T}_{\text{E}}\text{X}$ , Sopot (1991)
- A. Hoenig**, 1988, wykład prezentowany na Third Annual European  $\text{T}_{\text{E}}\text{X}$  Conference (Univ. Exeter, Anglia)
- D.E. Knuth**, 1990 (18-te wydanie), *The  $\text{T}_{\text{E}}\text{X}$ book*, Addison–Wesley Publ. Co., Reading (MA)
- D.E. Knuth**, 1986, *The  $\text{METAFont}$ -book*, Addison–Wesley Publ. Co., Reading (MA)
- J. Krieger, N. Schwarz**, 1990, *Introduction to  $\text{T}_{\text{E}}\text{X}$* , Addison–Wesley, Amsterdam
- L. Lamport**, 1985, *The  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  Document Preparation System*, Addison–Wesley Publ. Co., Reading (MA); tłum. polskie:  *$\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$*  (przekł. P. Wyrostek), ŚCO CYFRONET, Kraków
- M. Spivak**, 1985,  *$\text{PCT}_{\text{E}}\text{X}$  Manual*, Personal  $\text{T}_{\text{E}}\text{X}$ , Inc., Mill Valley (California)
- M.J. Wichura**, *The  $\text{P}_{\text{I}}\text{CT}_{\text{E}}\text{X}$  Manual*, The Univ. Chicago

**L**A<sub>T</sub>E<sub>X</sub> jest pakietem makrodefinicji znakomicie upraszczających korzystanie ze złożonego systemu składu drukarskiego T<sub>E</sub>X. System ten zdobył światowe uznanie głównie za sprawą wysokiej jakości składu przy darmowym udostępnianiu oprogramowania. W Polsce był dość ekskluzywnym narzędziem, ale własności najnowszej polskiej wersji T<sub>E</sub>X-a, tzw. M<sub>E</sub>X-a (opracowanego w spółce MacroSoft), rokuje znaczny wzrost popularności tego eksperta wśród edytorów.

**W** książce przedstawiono szczegółowo i przystępnie wszystkie najważniejsze polecenia L<sub>A</sub>T<sub>E</sub>X-a i wiele T<sub>E</sub>X-a. Od Czytelnika nie wymaga się żadnej uprzedniej znajomości tego systemu. Rozliczne przykłady ułatwią pierwsze praktyczne kroki, a dziesiątki zestawień tabelarycznych będą zawsze pomocne nawet dla zaawansowanych użytkowników.