‘Pots.tex’ and Other Useful Plain TEX Packages

Jacek Czekaj
University of Silesia, Institute of Mathematics
14 Bankowa st., 40-007 Katowice, Poland
jackens@math.us.edu.pl

Abstract

This article firstly describes the pots.tex package designed for plain TEX. This is a set of macros providing convenient methods for typesetting mathematical formulas with respect to old traditions and practice, with support for encoding Polish characters in UTF-8 standard. Subsequently, the article describes how to include graphics in extended TEX, and other useful packages designed for plain TEX, which are not well documented.

Introduction

Almost every mathematician has heard about TEX. Probably one can also say that most mathematicians greatly values TEX. But, as usual, there are some “dissatisfied”, for who something “does not fit” in TEX... The author of the article is also one of those who are “dissatisfied” and his “dissatisfaction” has arisen from the fact that as a mathematician, he has encountered with the former mathematical monographs, and noticed that some typographical “tricks” found there can not be easily achieved with TEX.

Therefore, the author wrote his own macro package pots.tex (designed for plain TEX) facilitates achievement of these tricks, and eliminates some disadvantages, such as requirement of writing up to four characters to get the letters ℓ in the math mode or lack of support for encoding Polish diacritics characters in UTF-8 standard in the plain TEX.

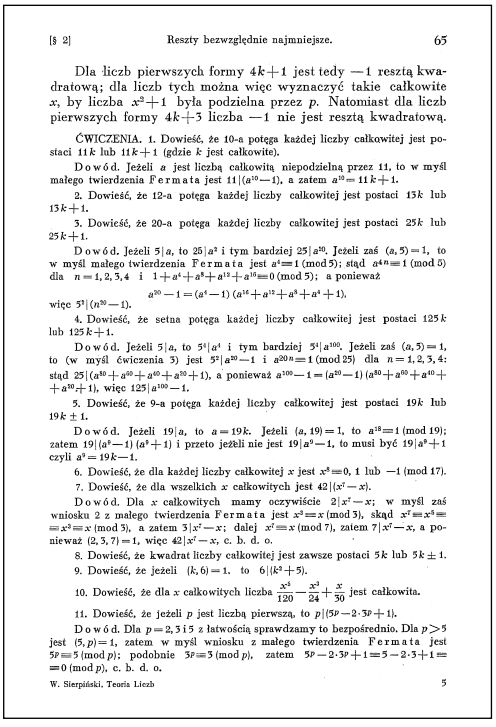
Macros of the ‘pots.tex’ package

Perhaps the macros of the package pots.tex are arranged in “unnatural” order... This unnatural order comes from the order in which they appear in the source file pots.tex and aims to help the exploration of sources of macros described here. (The sources are available at http://www.math.us.edu.pl/kcz/.)

☞ In the former Polish textbooks, breaking a mathematical formula at binary or relational symbol was associated with duplicating this symbol on the next line (conf. fig. 1):

Example formula with 4 pluses: $1+2+3+4+$
$+5=15$.

```
Example formula with 4 pluses:
$1+2+3+4+5=15$.
```



[§ 2] Reszty bezwzględnie najmniejsze. 65

Dla liczb pierwszych formy $4k+1$ jest tedy -1 resztą kwadratową; dla liczb tych można więc wyznaczyć takie całkowite x, by liczba x^2+1 była podzielna przez p. Natomiast dla liczb pierwszych formy $4k+3$ liczba -1 nie jest resztą kwadratową.

ĆWICZENIA. 1. Dowieść, że 10-a potęga każdej liczby całkowitej jest postaci $11k$ lub $11k+1$ (gdzie k jest całkowite).

Dowód. Jeżeli a jest liczbą całkowitą niepodzielną przez 11, to w myśl małego twierdzenia Fermata jest $11 \mid (a^{10}-1)$, a zatem $a^{10} = 11k+1$.

2. Dowieść, że 12-a potęga każdej liczby całkowitej jest postaci $13k$ lub $13k+1$.

3. Dowieść, że 20-a potęga każdej liczby całkowitej jest postaci $25k$ lub $25k+1$.

Dowód. Jeżeli $5 \mid a$, to $25 \mid a^2$ i tym bardziej $25 \mid a^{20}$. Jeżeli zaś $(a,5)=1$, to w myśl małego twierdzenia Fermata jest $a^4 \equiv 1 \pmod 5$; stąd $a^{4n} \equiv 1 \pmod 5$ dla $n=1,2,3,4$ i $1+a^4+a^8+a^{12}+a^{16} \equiv 0 \pmod 5$; a ponieważ

$$a^{20}-1 = (a^4-1)(a^{16}+a^{12}+a^8+a^4+1),$$

więc $5^3 \mid (a^{20}-1)$.

4. Dowieść, że setna potęga każdej liczby całkowitej jest postaci $125k$ lub $125k+1$.

Dowód. Jeżeli $5 \mid a$, to $5^4 \mid a^4$ i tym bardziej $5^4 \mid a^{100}$. Jeżeli zaś $(a,5)=1$, to (w myśl ćwiczenia 3) jest $5^2 \mid a^{20}-1$ i $a^{20n} \equiv 1 \pmod{25}$ dla $n=1,2,3,4$; stąd $25 \mid (a^{80}+a^{60}+a^{40}+a^{20}+1)$, a ponieważ $a^{100}-1 = (a^{20}-1)(a^{80}+a^{60}+a^{40}+a^{20}+1)$, więc $125 \mid a^{100}-1$.

5. Dowieść, że 9-a potęga każdej liczby całkowitej jest postaci $19k$ lub $19k \pm 1$.

Dowód. Jeżeli $19 \mid a$, to $a = 19k$. Jeżeli $(a,19)=1$, to $a^{18} \equiv 1 \pmod{19}$; zatem $19 \mid (a^9-1)(a^9+1)$ i przeto jeżeli nie jest $19 \mid a^9-1$, to musi być $19 \mid a^9+1$ czyli $a^9 = 19k-1$.

6. Dowieść, że dla każdej liczby całkowitej x jest $x^5 \equiv 0, 1$ lub $-1 \pmod{17}$.

7. Dowieść, że dla wszelkich x całkowitych jest $42 \mid (x^7-x)$.

Dowód. Dla x całkowitych mamy oczywiście $2 \mid x^7-x$; w myśl zaś wniosku 2 z małego twierdzenia Fermata jest $x^3 \equiv x \pmod 3$, skąd $x^7 \equiv x^5 \equiv x^3 \equiv x \pmod 3$, a zatem $3 \mid x^7-x$; dalej $x^7 \equiv x \pmod 7$, zatem $7 \mid x^7-x$, a ponieważ $(2,3,7)=1$, więc $42 \mid x^7-x$, c. b. d. o.

8. Dowieść, że kwadrat liczby całkowitej jest zawsze postaci $5k$ lub $5k \pm 1$.

9. Dowieść, że jeżeli $(k,6)=1$. to $6 \mid (k^2+5)$.

10. Dowieść, że dla x całkowitych liczba $\frac{x^5}{120} - \frac{x^3}{24} + \frac{x}{30}$ jest całkowita.

11. Dowieść, że jeżeli p jest liczbą pierwszą, to $p \mid (5p-2 \cdot 3^p+1)$.

Dowód. Dla $p=2, 3$ i 5 z łatwością sprawdzamy to bezpośrednio. Dla $p>5$ jest $(5,p)=1$, zatem w myśl wniosku z małego twierdzenia Fermata jest $5^p \equiv 5 \pmod p$; podobnie $3^p \equiv 3 \pmod p$, zatem $5^p - 2 \cdot 3^p + 1 \equiv 5 - 2 \cdot 3 + 1 \equiv 0 \pmod p$, c. b. d. o.

W. Sierpiński, Teoria Liczb 5

Fig. 1. Illustration of old-fashioned way of breaking mathematical formulas at binary or relational symbols on the example of mathematical monography by Wacław Sierpiński entitled “Teoria liczb” (Number theory) from the year 1950.

Furthermore, breaking formula at the subtraction operator should look as follows:

A simple example of a trivial equality: $10000 + -1000 = 1000$.

```
A simple example of a trivial equality:
$10000-1000=1000$.
```

(The equality holds in the binary system.)

In the package `pots.tex` breaking formula at binary or relational operator automatically inserts duplicated operator on the next line (or symbols $+$ $-$ in the case of a binary operator $-$).

☞ Symbols of divisibility relation and indivisibility relation have been changed:

$$2 \mid 6 \qquad 4 \nmid 6$$

```
$$2\mid6\qquad4\nmid6$$
```

☞ Commands providing italic version of Greek letters have been added:

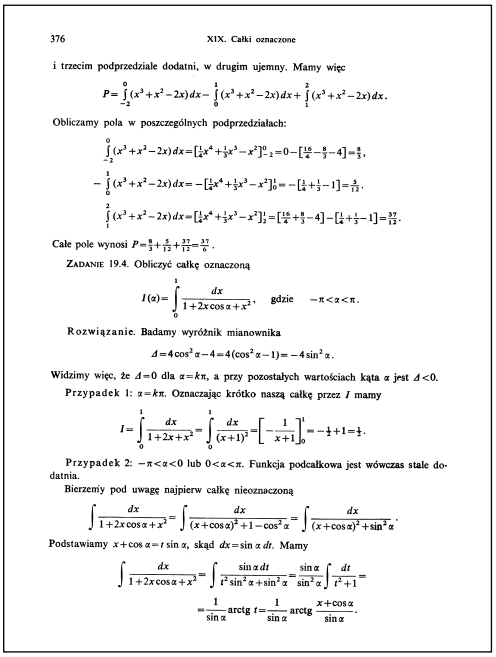
$$\alpha, \beta, \gamma, \delta, \epsilon, \zeta, \eta, \theta, \iota, \kappa, \lambda, \mu, \nu, \xi, \pi,$$
$$\rho, \sigma, \tau, \upsilon, \phi, \chi, \psi, \omega, \varepsilon, \vartheta, \varpi, \varrho, \varsigma, \varphi,$$
$$\Gamma, \Delta, \Theta, \Lambda, \Xi, \Pi, \Sigma, \Upsilon, \Phi, \Psi, \Omega$$

```
$$\displaylines{
  \italpha,\itbeta,\itgamma,\itdelta,
  \itvarepsilon,\itzeta,\iteta,
  \itttheta,\itiota,\itkappa,\itlambda,
  \itmu,\itnu,\itxi,\itpi,\cr
  \itrho,\itsigma,\ittau,\itupsilon,
  \itvarphi,\itchi,\itpsi,\itomega,
  \itepsilon,\itvartheta,\itvarpi,
  \itvarrho,\itvarsigma,\itphi,\cr
  \itGamma,\itDelta,\itTheta,\itLambda,
  \itXi,\itPi,\itSigma,\itUpsilon,
  \itPhi,\itPsi,\itOmega}$$
```

☞ In the Polish typography the size of the integration symbol should depend on the integrated function, rather than display math mode or text mode (conf. fig. 2):

$$\int_a^b f(x)/g(x)\,dx = \int_a^b \frac{f(x)}{g(x)}dx$$

```
$$\int_a^bf(x)/g(x)\,dx=
  \Int_a^b\Frac{f(x)}{g(x)}dx$$
```



376 XIX. Całki oznaczone

i trzecim podprzedziale dodatni, w drugim ujemny. Mamy więc

$$P=\int_{-2}^{0}(x^3+x^2-2x)\,dx-\int_{0}^{1}(x^3+x^2-2x)\,dx+\int_{1}^{2}(x^3+x^2-2x)\,dx.$$

Obliczamy pola w poszczególnych podprzedziałach:

$$\int_{-2}^{0}(x^3+x^2-2x)\,dx=\left[\tfrac{1}{4}x^4+\tfrac{1}{3}x^3-x^2\right]_{-2}^{0}=0-\left[\tfrac{16}{4}-\tfrac{8}{3}-4\right]=\tfrac{8}{3},$$

$$-\int_{0}^{1}(x^3+x^2-2x)\,dx=-\left[\tfrac{1}{4}x^4+\tfrac{1}{3}x^3-x^2\right]_{0}^{1}=-\left[\tfrac{1}{4}+\tfrac{1}{3}-1\right]=\tfrac{5}{12},$$

$$\int_{1}^{2}(x^3+x^2-2x)\,dx=\left[\tfrac{1}{4}x^4+\tfrac{1}{3}x^3-x^2\right]_{1}^{2}=\left[\tfrac{16}{4}+\tfrac{8}{3}-4\right]-\left[\tfrac{1}{4}+\tfrac{1}{3}-1\right]=\tfrac{37}{12}.$$

Całe pole wynosi $P=\frac{8}{3}+\frac{5}{12}+\frac{37}{12}=\frac{37}{6}$.

ZADANIE 19.4. Obliczyć całkę oznaczoną

$$I(\alpha)=\int_0^1\frac{dx}{1+2x\cos\alpha+x^2},\quad \text{gdzie}\quad -\pi<\alpha<\pi.$$

Rozwiązanie. Badamy wyróżnik mianownika

$$\Delta=4\cos^2\alpha-4=4(\cos^2\alpha-1)=-4\sin^2\alpha.$$

Widzimy więc, że $\Delta=0$ dla $\alpha=k\pi$, a przy pozostałych wartościach kąta α jest $\Delta<0$.

Przypadek 1: $\alpha=k\pi$. Oznaczając krótko naszą całkę przez I mamy

$$I=\int_0^1\frac{dx}{1+2x+x^2}=\int_0^1\frac{dx}{(x+1)^2}=\left[-\frac{1}{x+1}\right]_0^1=-\tfrac{1}{2}+1=\tfrac{1}{2}.$$

Przypadek 2: $-\pi<\alpha<0$ lub $0<\alpha<\pi$. Funkcja podcałkowa jest wówczas stale dodatnia.

Bierzemy pod uwagę najpierw całkę nieoznaczoną

$$\int\frac{dx}{1+2x\cos\alpha+x^2}=\int\frac{dx}{(x+\cos\alpha)^2+1-\cos^2\alpha}=\int\frac{dx}{(x+\cos\alpha)^2+\sin^2\alpha}.$$

Podstawiamy $x+\cos\alpha=t\sin\alpha$, skąd $dx=\sin\alpha\,dt$. Mamy

$$\int\frac{dx}{1+2x\cos\alpha+x^2}=\int\frac{\sin\alpha\,dt}{t^2\sin^2\alpha+\sin^2\alpha}=\frac{\sin\alpha}{\sin^2\alpha}\int\frac{dt}{t^2+1}=$$
$$=\frac{1}{\sin\alpha}\operatorname{arctg}t=\frac{1}{\sin\alpha}\operatorname{arctg}\frac{x+\cos\alpha}{\sin\alpha}.$$

Fig. 2. Ilustration of old-fashioned way of selecting the size of the integration symbol depending on the sub-integral function on the example of mathematical monography by Włodzimierz Krysicki and Lech Włodarski entitled "Analiza matematyczna w zadaniach. Część 1" (Mathematical Analysis by Example. Part 1).

These operators symbols are always the same size regardless of text mode or display math mode:

$$\wedge = \wedge, \bigwedge = \bigwedge, \forall, \vee = \vee, \bigvee = \bigvee, \exists,$$
$$\amalg, \coprod, \Sigma, \sum, \Pi, \prod, \int, \int, \oint, \oint, \cap, \bigcap,$$
$$\cup, \bigcup, \sqcup, \bigsqcup, \odot, \bigodot, \otimes, \bigotimes, \oplus, \bigoplus, \uplus, \biguplus$$

```
$$\displaylines{
  \forall=\bigwedge,\Forall=\Bigwedge,
  \plainforall,\exists=\bigvee,
  \Exists=\Bigvee,\plainexists,\cr
  \coprod,\Coprod,\sum,\Sum,\prod,
  \Prod,\int,\Int,\oint,\Oint,\bigcap,
  \Bigcap,\cr
  \bigcup,\Bigcup,\bigsqcup,\Bigsqcup,
  \bigodot,\Bigodot,\bigotimes,
  \Bigotimes,\bigoplus,\Bigoplus,
  \biguplus,\Biguplus}$$
```

☞ Regardless of the text mode or display math mode indices of all operator symbols are typeseted over and under the symbol by default, rather than beside of the symbol. To force plain TEX's behavior a command `\nolimits` is required:

$$\int_a^b \frac{1}{f(x)}\,dx + \lim_{n\to\infty}\frac{1}{n} = \int_a^b \frac{1}{f(x)}\,dx + \lim\nolimits_{n\to\infty}\frac{1}{n}$$

```
$$\Int_a^b\Frac1{f(x)}\,dx+
  \lim_@{n\to\infty}\frac1n=
  \Int\nolimits_a^b\Frac1{f(x)}\,dx+
  \lim\nolimits_{n\to\infty}\frac1n$$
```

The same applies to the following operators:

$$\inf, \lim, \liminf, \limsup, \max, \min, \sup$$

```
$$\inf,\lim,\liminf,\limsup,\max,\min,
  \sup$$
```

While the indices of other operators are typeseted beside of them:

$$\sin^2\alpha = (\sin\alpha)^2 \qquad \lg = \log_2$$

```
$$\sin^2\alpha=(\sin\alpha)^2\qquad
  \lg=\log_2$$
```

This applies to operators:

$$\arccos, \arcsin, \arctan, \cos, \cosh, \cot, \coth, \csc, \deg, \det, \dim, \exp, \hom, \ker, \lg, \ln, \log, \ord, \rank, \sec, \sin, \sinh, \tan, \tanh, \mathrm{NWD}, \mathrm{NWW}, \Pr$$

```
$$\displaylines{
  \arccos,\arcsin,\arctan,\cos,\cosh,
  \cot,\coth,\csc,\cr
  \deg,\det,\dim,\exp,\hom,\ker,\lg,
  \ln,\log,\ord,\rank,\cr
  \sec,\sin,\sinh,\tan,\tanh,
  \NWD,\NWW,\Pr}$$
```

☞ Height of the brackets is automatically adjusted to the size of "material" between them:

$$\left[\frac{1}{2},\frac{3}{4}\right) = [\frac{1}{2},\frac{3}{4})$$

```
$$[\Frac12,\Frac34)=
 \[\Frac12,\Frac34\)$$
```

This applies to the following characters and commands:

$$(, [, \langle, \lceil, \lfloor,], \rceil, \rangle,],)$$

```
$$(,[,\<,\lceil,\lfloor,
  \vrule width0ptheight2ex
  \rfloor,\rceil,\>,],)$$
```

Instead of `\<` and `\>` commands `\langle` and `\rangle` can be used. Symbols typesetted without an automatic adjustment can be obtained as follows:

$$(, [, \langle, \lceil, \lfloor,], \rceil, \rangle,],)$$

```
$$\(,\[,\plainlangle,\plainlceil,
  \plainlfloor,\plainrfloor,
  \plainrceil,\plainrangle,\],\)$$
```

☞ Default plain TEX's behavior in the case of breaking mathematical formulas after comma or semicolon have been changed:

How many elements is in set: $\{a,b,c,d,e,f,g,h,i,j,k,\ell,m,n,o,p,q,r,s,t,u,v,w,x,y,z\}$?

```
How many elements is in set:
$\{a,b,c,d,e,f,g,h,i,j,k,l,m,
   n,o,p,q,r,s,t,u,v,w,x,y,z\}$?
```

(The set contains at most 26 elements...)

☞ In the math mode * produces the same character as \cdot command:

$$a \cdot b = b \cdot a$$

```
$$a*b=b\cdot a$$
```

There is no need to use commands \cdots and \dots, since * and . are enough:

$$1 + \cdots + n = \frac{n(n+1)}{2} \qquad \text{for} \quad n = 1, 2, \ldots$$

```
$$1+****+n=\frac{n(n+1)}2\qquad
  \hbox{for}\quad n=1,2,...$$
```

☞ For convenience, in the package pots.tex, characters obtained with command \epsilon have been changed:

$$\varepsilon = \varepsilon \neq \epsilon$$

```
$$\epsilon=\varepsilon\ne\plainepsilon$$
```

Similarly, in the case of letter phi:

$$\varphi = \varphi \neq \phi$$

```
$$\phi=\varphi\ne\plainphi$$
```

☞ Also for convenience the shape of the letter "l" in the math mode have been changed:

$$\ell = \ell \neq l$$

```
$$l=\ell\ne\plainl$$
```

☞ In math mode character @ has a special meaning, it causes typeset the next character (or a group of characters) in the \scriptstyle mode in box of zero width and centered content. This allows typesetting conveniently supscripts and subscripts of operators:

$$\sum_{i=1}^{100000} a_i + \sum_{i=100001}^{200000} a_i = \sum_{i=1}^{100000} a_i + \sum_{i=100001}^{200000} a_i$$

```
$$\Sum_@{i=1}^@{100000}a_i+
  \Sum_@{i=100001}^@{200000}a_i=
  \Sum_{i=1}^{100000}a_i+
  \Sum_{i=100001}^{200000}a_i$$
```

☞ There is no need to use command \colon which is known from plain TEX, since in math mode a colon behaves just like \colon (inserts a colon with a special space only on the right):

$$f\colon X \to Y \quad\Longleftrightarrow\quad f\colon X \to Y$$

```
$$f:X\to Y\quad\iff\quad
  f\colon X\to Y$$
```

☞ Macro \widehat omitted in plain TEX format have been added:

$$\widehat{abc}$$

```
$$\widehat{abc}$$
```

☞ Characters "less than or equal to" and "greater than or equal to" are typeseted with respect to Polish practice:

$$\leqslant = \leqslant = \leqslant = \leqslant \neq \leq \qquad \geqslant = \geqslant = \geqslant = \geqslant \neq \geq$$

```
$${\le}={\leq}={\xle}={\xleq}
  \ne{\plainle}\qquad
  {\ge}={\geq}={\xge}={\xgeq}
  \ne{\plainge}$$
```

☞ Characters obtined with \equiv and \cong have been exchanged:

$$\equiv = \equiv \neq \cong$$

```
$${\cong}={\equiv}\ne{\plaincong}$$
```

☞ Command `\iff` do not insert an extra space around the symbol of equivalence (the command works just like `\Longleftrightarrow` command):

$$a=b \iff b=a \qquad a=b \iff b=a$$

```
$$a=b\iff b=a\qquad
  a=b\Longleftrightarrow b=a$$
```

☞ Commands `\<` and `\>` provides convenient typesetting angled brackets have been added:

$$\left\langle \frac{1}{2}, \frac{3}{4} \right\rangle = \left\langle \frac{1}{2}, \frac{3}{4} \right\rangle$$

```
$$\<\Frac12,\Frac34\>=
  \langle\Frac12,\Frac34\rangle$$
```

☞ Macro `\matrix` do not insert too large gaps around the matrix, so that the brackets surrounding matrix looks more aesthetically:

$$\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$$

```
$$\det[\matrix{a&b\cr c&d}]=ad-bc$$
```

☞ Macro `\eqalign` has been improved by the possibility of typeset many aligned columns, and not just two as in plain TEX:

$$\begin{aligned} 1 \cdot 1 &= 1, & 2 + 2 &= 4, & 1 + 2 &= 3, \\ 1 + 1 &= 2, & 2 \cdot 2 &= 4, & 1 + 1 + 2 &= 4. \end{aligned}$$

```
$$\eqalign{1*1&=1,&\qquad
           2+2&=4,&
           1+2&=3,\cr
           1+1&=2,&
           2*2&=4,&\qquad
         1+1+2&=4.}$$
```

☞ Convenient shortcuts for symbols of: the set of primes, the set of natural numbers, the set of integers, the set of rational numbers, the set of real numbers and the set of complex numbers have been added:

$$\mathbf{P} \subseteq \mathbf{N} \subseteq \mathbf{Z} \subseteq \mathbf{Q} \subseteq \mathbf{R} \subseteq \mathbf{C}$$

```
$$\PP\subset\NN\subset\ZZ\subset
  \QQ\subset\RR\subset\CC$$
```

Originally in Polish literature the bold font was used to denote those symbols, rather than one that simulates the appearance of those signs written with chalk on a blackboard!

☞ Macro `\mod` provides typeset conveniently congruences with respect to Polish practice:

$$10 \equiv 1 \pmod 3$$

```
$$10\cong1\mod3$$
```

☞ Fraction's sizes, as well as operators symbols' sizes, do not depend on text mode or displayed math mode, but should be selected depending on the contents of the numerator and denominator:

$$\tfrac{1}{2} = \frac{1}{2}$$

```
$$\frac12=\Frac12$$
```

☞ Macro `\intertext` providing inserting text between aligned formulas easily have been added:

$$\begin{aligned} 1 + 1 &= 2 && (1) \\ &\text{and} \\ 1 + 1 + 1 &= 3 && (2) \end{aligned}$$

```
$$\eqalignno{1+1&=2    &(1)\cr
   \intertext{and}
         1+1+1&=3    &(2)}$$
```

☞ Macros `\newpage` and `\\` well known from the LATEX format are available:

Do not break the text manualy
just leave it TEX...

```
Do not break the text manualy\\
just leave it \TeX\dots
```

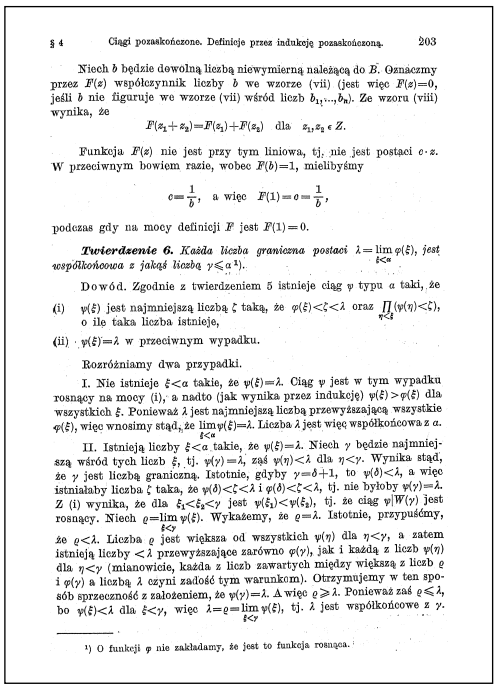
☞ Macro providing “hiding” operator’s indices in formulas with operators with respect to Polish typographical practice have been added (conf. fig. 3):

This paragraph with sum $\sum_{i=1}^{n} a_i$ looks better than the following.

This paragraph with sum $\sum_{i=1}^{n} a_i$ looks worse than the above.

```
This paragraph with sum
$\smashop{\sum_@{i=1}}^na_i$
looks better than the following.

This paragraph with sum
$\sum_@{i=1}^na_i$
looks worse than the above.
```



§ 4 Ciągi pozaskończone. Definicje przez indukcję pozaskończoną. 203

Niech b będzie dowolną liczbą niewymierną należącą do B. Oznaczmy przez $F(z)$ współczynnik liczby b we wzorze (vii) (jest więc $F(z)=0$, jeśli b nie figuruje we wzorze (vii) wśród liczb $b_1,\ldots,b_n$). Ze wzoru (viii) wynika, że

$$F(z_1+z_2)=F(z_1)+F(z_2) \quad \text{dla} \quad z_1,z_2 \in Z.$$

Funkcja $F(z)$ nie jest przy tym liniowa, tj. nie jest postaci $c\cdot z$. W przeciwnym bowiem razie, wobec $F(b)=1$, mielibyśmy

$$c=\frac{1}{b}, \quad \text{a więc} \quad F(1)=c=\frac{1}{b},$$

podczas gdy na mocy definicji F jest $F(1)=0$.

Twierdzenie 6. *Każda liczba graniczna postaci* $\lambda=\lim_{\xi<\alpha}\varphi(\xi)$, *jest współkońcowa z jakąś liczbą* $\gamma\leqslant\alpha$ [1]).

Dowód. Zgodnie z twierdzeniem 5 istnieje ciąg ψ typu α taki, że

(i) $\psi(\xi)$ jest najmniejszą liczbą ζ taką, że $\varphi(\xi)<\zeta<\lambda$ oraz $\prod_{\eta<\xi}(\psi(\eta)<\zeta)$, o ile taka liczba istnieje,

(ii) $\psi(\xi)=\lambda$ w przeciwnym wypadku.

Rozróżniamy dwa przypadki.

I. Nie istnieje $\xi<\alpha$ takie, że $\psi(\xi)=\lambda$. Ciąg ψ jest w tym wypadku rosnący na mocy (i), a nadto (jak wynika przez indukcję) $\psi(\xi)>\varphi(\xi)$ dla wszystkich ξ. Ponieważ λ jest najmniejszą liczbą przewyższającą wszystkie $\varphi(\xi)$, więc wnosimy stąd, że $\lim_{\xi<\alpha}\psi(\xi)=\lambda$. Liczba λ jest więc współkońcowa z α.

II. Istnieją liczby $\xi<\alpha$ takie, że $\psi(\xi)=\lambda$. Niech γ będzie najmniejszą wśród tych liczb ξ, tj. $\psi(\gamma)=\lambda$, zaś $\psi(\eta)<\lambda$ dla $\eta<\gamma$. Wynika stąd, że γ jest liczbą graniczną. Istotnie, gdyby $\gamma=\delta+1$, to $\psi(\delta)<\lambda$, a więc istniałaby liczba ζ taka, że $\psi(\delta)<\zeta<\lambda$ i $\varphi(\delta)<\zeta<\lambda$, tj. nie byłoby $\psi(\gamma)=\lambda$. Z (i) wynika, że dla $\xi_1<\xi_2<\gamma$ jest $\psi(\xi_1)<\psi(\xi_2)$, tj. że ciąg $\psi|W(\gamma)$ jest rosnący. Niech $\varrho=\lim_{\xi<\gamma}\psi(\xi)$. Wykażemy, że $\varrho=\lambda$. Istotnie, przypuśćmy, że $\varrho<\lambda$. Liczba ϱ jest większa od wszystkich $\psi(\eta)$ dla $\eta<\gamma$, a zatem istnieją liczby $<\lambda$ przewyższające zarówno $\varphi(\gamma)$, jak i każdą z liczb $\psi(\eta)$ dla $\eta<\gamma$ (mianowicie, każda z liczb zawartych między większą z liczb ϱ i $\varphi(\gamma)$ a liczbą λ czyni zadość tym warunkom). Otrzymujemy w ten sposób sprzeczność z założeniem, że $\psi(\gamma)=\lambda$. A więc $\varrho\geqslant\lambda$. Ponieważ zaś $\varrho\leqslant\lambda$, bo $\psi(\xi)<\lambda$ dla $\xi<\gamma$, więc $\lambda=\varrho=\lim_{\xi<\gamma}\psi(\xi)$, tj. λ jest współkońcowe z γ.

[1]) O funkcji φ nie zakładamy, że jest to funkcja rosnąca.

Fig. 3. Illustration of the typographic effect of “hiding operator’s indices” on example of mathematical monography by Kazimierz Kuratowski and Andrzej Mostowski entitled “Teoria mnogości” (Set Theory) from year 1952.

☞ Command `\qed` inserting symbol □ at the end of the line, often placed at the end of the proof have been introduced:

Theorem 1. For each number $n \in \mathbf{N}$ hold equality

$$1+\cdots+n=\frac{n\,(n+1)}{2}.$$

Proof. The proof is left the reader as an exercise. □

```
{\bf Theorem 1.} \ For each number
$n\in\NN$ hold equality
$$1+***+n=\Frac{n(n+1)}2.$$
{\it Proof.} \ The proof is left
the reader as an exercise.\qed
```

The symbol can be also placed in displayed formulas:

Proof. The proof follows from the equalities:

$$1+\cdots+n=\sum_{i=1}^{n} i=\frac{n\,(n+1)}{2}. \qquad \square$$

```
{\it Proof.} \ The proof follows
from the equalities:
$$1+***+n=\sum_@{i=1}^ni=
  \Frac{n(n+1)}2.\eqno\qed$$
```

As well as in aligned displeyed formulas:

Proof. The proof follows from the equalities:

$$\begin{aligned} 1+\cdots+n &= \frac{(1+\cdots+n)+(n+\cdots+1)}{2} = \\ &= \frac{(n+1)+\cdots+(n+1)}{2} = \\ &= \frac{n\,(n+1)}{2}. \end{aligned} \qquad \square$$

```
{\it Proof.} \ The proof follows
from the equalities:
$$\eqalignno{1+***+n&=
  \Frac{(1+***+n)+(n+***+1)}2=\cr&=
  \Frac{(n+1)+***+(n+1)}2=\cr&=
  \Frac{n(n+1)}2.&\qed}$$
```

☞ Commands `\x`, `\xx` and `\xxx` providing typesetting “star” signed formulas easily have been added:

First trivial equality:
$$1=1. \eqno(*)$$
Second trivial equality:
$$1+1=2. \eqno(**)$$
Third trivial equality:
$$1+1+1=3. \eqno(***)$$

```
First trivial equality:
$$1=1.\eqno\x$$
Second trivial equality:
$$1+1=2.\eqno\xx$$
Third trivial equality:
$$1+1+1=3.\eqno\xxx$$
```

☞ Typesetting sets in American notation i.e. with vertical bar instead of Polish colon have been added. Macro `\:` inserts appropriate gap on the right side of the symbol | .

$$\{n \in \mathbf{N} | \; n^2 \notin \mathbf{N}\} = \emptyset$$

```
$$\{n\in\NN\:n^2\notin\NN\}=\emptyset$$
```

☞ Macros providing typesetting pseudocodes easily have been added:

GCD(a, b)
1: **while** b $\neq$ 0 **do**
2: set c = a (mod b)
3: set a = b
4: set b = c
5: **return** a

```
\pseudocode
GCD$(a,b)$
while $b\ne0$ do
    "set" $c=a\mod b$
    "set" $a=b$
    "set" $b=c$
return $a$
\endpseudocode
```

"Indentations" of the pseudocode lines are obtained by tabulators or regular spaces. Width of the gap for line numbers can be changed with dimen `\tabwidth` (it is equal to `\parindent` by default). Macros `\linenumbers` and `\nolinenumbers` allows enabling and disabling automatic line numbering.

☞ Macro `\insertpdfpage` allows including any page selected from the specified `pdf` file. For example, command

```
\insertpdfpage3{The_TeXbook.pdf}
```

would insert third page from file `The_TeXbook.pdf`.

☞ Package `pots.tex` supports typesetting Polish texts with Polish diacritics characters encoded in UTF-8 standard.

Including graphics

☞ Extended TEX supports including images in formats `pdf`, `png` and `jpeg` directly:


```
$$\pdfximage width2cm{horTeX.png}
  \pdfrefximage\pdflastximage$$
```

If only height of the image is specified, the width will be automatically properly selected, with respect to the image aspect ratio. Similarly, when you specify only the width. Specifying explicitly width or height of the image is not necessary. In this case, the image will have its own "natural" size. Specifing both the width and the height of the image is also correct. In the case of the format `pdf` specifing included page number is also allowed, for example command:

```
\pdfximage page3{The_TeXbook.pdf}
\pdfrefximage\pdflastximage
```

would insert third page from file `The_TeXbook.pdf` (macro `\insertpdfpage` inserts page selected from the `pdf` file as a separate page).

If re-insert of the same image is needed, its number should been saved:

```
\edef\horTeX{\the\pdflastximage}
```

to use it without unnecessarily multiplied including image file to the resulting `pdf` file:


```
$$\pdfrefximage\horTeX\quad
  \pdfrefximage\horTeX\quad
  \pdfrefximage\horTeX$$
```

☞ Extended TEX provides direct `pdf` tags insertion, for example we can change a color of the text:

Buy rose-tinted glasses, even in black market...

```
\pdfliteral{.9 .1 .7 rg}%
Buy rose-tinted glasses,
\pdfliteral{0 0 0 rg}%
even in black market\dots
```

Three numbers in the range from 0 to 1 determines red, green and blue component, respectively.

Other useful plain TEX's packages

The list of useful plain TEX's packages should not omit such packages as:

☞ `tap.tex` — typesetting tables conveniently;

☞ `tikz.tex` — making graphics (charts, graphs, etc.);

☞ `pdf-trans.tex` — box transformations and frames (eg. frames in this paper);

☞ `tp-crf.tex` — cross references.

The above packages are well documented, so we do not describe them here. We focus on packages less well known and less well documented.

☞ Package `multicol.tex` allows typesetting multicolumn. Columns are automatically balances, as required by Polish typography. Command `\columnsep` = *dimen* determines the gap between columns, command `\columnseprule` = = *dimen* determines the width of the optional vertical line between columns, and command `\multicolbaselineskip` = *dimen* allows specifing an optional interline correction, i.e. the value of `\baselineskip` parameter.

Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua.

Ut enim ad minima veniam, quis nostru exercitation ullam co laborios, nisi ut aliquid ex ea commodi consequat.

```
\beginmulticols2
Lorem ipsum dolor sit, amet,
consectetur, adipisci elit, sed eius
mod tempor incidunt, ut labore et
dolore magna aliqua. \
Ut enim ad minima veniam, quis nostru
exercitation ullam co laborios, nisi
ut aliquid ex ea commodi consequat.
\endmulticols
```

☞ Package `verbatim-dek.tex` or `verbatim.tex` in older TEX's distributions (eg. TEXLive 2008) provides typesetting "verbatim" texts, for example as in all blue frames in this article:

```
"Verbatim" text: @#$%&#...
```

```
\let\verb=\verbatim
\verb!"Verbatim" text: @#$%&#...!
```

For convenience, we shortened to long command `\verbatim` to much nicer `\verb`. The text between the first mark after command `\verb`, and the second same mark are typesetted verbatim. (In the above example, this mark is an exclamation.)

☞ Package `insbox.tex` provides various inserts (eg. illustrations):

Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua. Lorem  ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua.
Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut  labore et dolore magna aliqua. Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua.
Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua.
Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua. 

Lorem ipsum dolor sit, amet, consectetur, adipisci elit, sed eius mod tempor incidunt, ut labore et dolore magna aliqua.

```
\def\material{\pdfliteral{0 .9 0 rg}%
\def\ins{\pdfliteral{0 .9 0 rg}%
  \vrule width1cmheight1cm
  \pdfliteral{0 0 0 rg}}
\def\lorem{Lorem ipsum dolor sit, amet,
  consectetur, adipisci elit, sed eius
  mod tempor incidunt, ut labore et
  dolore magna aliqua. }
\InsertBoxR2{\hbox to1cm{\kern-16pt
  \ins}}[-1]\lorem\lorem
\par\InsertBoxL2{\ins}[-1]\lorem\lorem
\par\InsertBoxC{\ins}\lorem
\par\InsertBoxR1{\hbox to1cm{\kern-16pt
  \ins}}\lorem
\par\MoveBelowBox\lorem
```

☞ Package `antt-math.tex` switches font family to Antykwa Toruńska. Package `iwona-math.tex` switches font family to Iwona (Yvonne), while the package `courier-math.tex` switches font family to Kurier. For example, Iwona font family presents as follows:

The quick brown fox jumps over the lazy dog 123
The quick brown fox jumps over the lazy dog 123
The quick brown fox jumps over the lazy dog 123

```
\tenpoint The quick brown fox
  jumps over the lazy dog 123
\it The quick brown fox
  jumps over the lazy dog 123
\bf The quick brown fox
  jumps over the lazy dog 123
```

Each of these packages provides commands `\eightpoint`, `\ninepoint`, `\tenpoint`, `\elevenpoint`, `\twelvepoint`, `\fourteenpoint`, `\seventeenpoint`, used to select font size.

☞ Package `cap.tex` provides typesetting source codes of C-like and Pascal-like languages easily:

```
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
  cout << "Witaj swiecie!" << endl;
//cout << "Hello, world!"  << endl;
  return 0;
}
```

```
\BeginC
#include <iostream>
using namespace std;
int main(int argc, char** argv) {
  cout << "Witaj swiecie!" << endl;
//cout << "Hello, world!"  << endl;
  return 0;
}
\EndC
```

```
program Hello_world;
begin
  writeln('Witaj swiecie!');
{ writeln('Hello, world!'); }
end.
```

```
\BeginPascal
program Hello_world;
begin
  writeln('Witaj swiecie!');
{ writeln('Hello, world!'); }
end.
\EndPascal
```

Including source code from an external file is also possible, eg.:

```
\InputC{dijkstra.cpp}
```

will typseset the content of file `dijkstra.cpp`, while

```
\InputPascal{dijkstra.pas}
```

will typseset the content of file `dijkstra.pas`.

☞ Package `underlin.tex` provides aesthetic text underlining:

Aesthetically underlined text...
Text underlined with plain TEX's macros...

```
\textul{Aesthetically underlined
  text\dots}\\
$\underline{\hbox{Text underlined with
  plain \TeX's macros\dots}}$
```

☞ Package `wiggly.tex` provides aesthetic "wiggly" text underlining:

"Wiggly" underlined text...

```
\underwiggle{``Wiggly''
  underlined text\dots}
```

☞ Including packages `map.tex`, `split.tex` and `tsp.tex` allows typesetting spaced out text:

A l i c e h a s a c a t . . .

```
\def\tspsp{0.9em plus.3em minus.2em}%
\def\tsplet{.1666667em}%
\tsp{Alice has a cat\dots}
```

Including packages `map.tex`, `split.tex` and `tun.tex` allows to undeline or strike out some text:

Alice has a cat... Alice has a cat...
~~Alice has a cat...~~ ~~Alice has a cat...~~

```
\def\tunsep{.25ex}\def\tunwd{.5pt}%
\tun{Alice has a cat\dots}
\tun{Alice\ has\ a\ cat\dots}\\
\def\tunsep{-.7ex}\def\tunwd{1pt}%
\tun{Alice has a cat\dots}
\tun{Alice\ has\ a\ cat\dots}
```

☞ Package `compare.tex` provides comparing strings lexicographically:

-1, 1, 1, 0

```
\compare{aa}{ab}, \compare{ab}{aa},
\compare{aa}{Ab}, \compare{aa}{aa}
```

☞ Package `binhex.tex` provides automate conversions of decimal numbers to binary and hexadecimal:

10010011001011000000001011010010, 499602D2

```
\binary{1234567890}, \hex{1234567890}
```

☞ Package `licz.mex` provides automate conversion numbers to the Polish verbal form:

jeden miliard dwieście trzydzieści cztery miliony pięćset sześćdziesiąt siedem tysięcy osiemset dziewięćdziesiąt

```
\licz{1234567890}
```

Summary

As we all know there is no accounting for taste... Probably there are many of those who would "subscribe to" other principles of typesetting mathematics, rather than presented in this paper. The author does not want to convince anyone to principles presented here. This work intends only to present that the plain TEX format, or MEX in Polish, is still "alive" and a small amount of work allows to reach any typographic "effects" well known from the lecture of old academic manusctipts.