

V. Matematyka w \LaTeX -u

Wiesław Krakowiak

16 grudnia 2012

Składanie formuł matematycznych w \LaTeX -u

W czystym \TeX -u są polecenia pozwalające poprawnie składać teksty matematyczne. W \LaTeX -u składanie formuł matematycznych uległo uproszczeniu. Został wzbogacony o polecenia pozwalające w sposób prostszy składać skomplikowane formuły. Zadaniem autora jest przekazanie jedynie logicznej struktury formuły, jej skład wykona samodzielnie \LaTeX .

Nie należy z reguły używać matematycznych poleceń \TeX -a w \LaTeX -u, gdyż powoduje to błędy przy kompilacji.

W \LaTeX -u można również ponumerować formuły matematyczne oraz w sposób prosty powoływać się na nie.

Składanie formuł matematycznych w \LaTeX -u (cd)

Jeżeli w pisanym tekście matematycznym używamy nieskomplikowanej matematyki, to do składania wystarczą jedynie polecenia „czystego” \LaTeX -a. W przeciwnym przypadku należy zainstalować dodatkowe pakiety \mathcal{AMS} - \LaTeX .

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$

Pod patronatem Amerykańskiego Towarzystwa Matematycznego (American Mathematical Society) został w roku 1982 stworzony pakiet $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ będący rozszerzeniem $\mathcal{T}\mathcal{E}\mathcal{X}$ -a. Uczynił on otrzymywane dokumenty bardziej spójnymi jak również przyspieszył czas ich generacji. Doprowadziło to w konsekwencji do wykreowania z tego rozszerzenia serii pakietów o wspólnej nazwie $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$. Ponieważ czcionki zaimplementowane w $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ -u muszą korespondować z $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -em doprowadziło to stworzenia osobnej grupy pakietów o nazwie `amsfonts`. Podstawowa zaleta jest ułatwienie składania tekstów matematycznych dając plik spełniający najwyższe standardy wydawnictw matematycznych.

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}(\text{cd})$

Pakiet $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ dostarcza serii predefiniowanych poleceń (`nmatrix`, `ntext`) które znacznie usprawniają posługiwanie się środowiskiem matematycznym poprzez wdrożenie doświadczeń standardów $\mathcal{A}\mathcal{M}\mathcal{S}$ i radzenie sobie ze złożonym i problemami takimi jak macierz wewnątrz macierzy lub słowa w indeksach dolnych i górnych bez obciążania tym użytkownika. Projekt $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ pojawił się w 1987 r, 3 lata później została wypuszczona pierwsza wersja tego programu. Przekształcenie $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{T}\mathcal{E}\mathcal{X}$ -a jego matematycznymi możliwościami do $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a zostało dokonane przez Franka Mittelbacha i Ranera Schopfa.

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}(\text{cd})$

$\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ składa się z różnych rozszerzeń $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a i jest on częścią jego standardowej dystrybucji. Podzielony jest na dwie części:

- `amsc1s` – dzięki temu pakietowi dokumenty w $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ wyglądać mają jak dokumenty $\mathcal{A}\mathcal{M}\mathcal{S}$.
- `amsmath` – ułatwia pisanie formuł matematycznych i ulepsza sposób ich prezentacji

Pakiet `amsmath` wprowadza wiele usprawnień, których zadaniem jest polepszenie prezentacji matematycznych formuł w dokumencie. W jego skład wchodzi kilka pomocniczych pakietów: **`amsmath`**, **`amstext`**, **`amsopn`**, **`amsbsy`**, **`amscd`**, oraz **`amsxtra`**.

Przegląd pakietów $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a

amsbsy definiuje polecenia `\boldsymbol` i `\pmb`, które udostępniają dwa rodzaje pogrubienia symboli matematycznych.

amscd pakiet ten jest niezależny od pozostałych pakietów; można go używać nie deklarując innych. Definiuje on otoczenie `CD` i wiele poleceń przydatnych do tworzenia diagramów przemiennych. Pakiet nie generuje strzałem skierowanych ukośnie.

amsfonts pakiet umożliwia korzystanie z poleceń `\mathbb` i `\mathfrak`, służących do pisania literami „tablicowymi” i gotyckimi”. Pakiet ten jest wczytywany automatycznie po zadeklarowaniu pakietu **amssymb**.

Przegląd pakietów $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a (cd)

amsgen pełni rolę pomocniczą, wspomagając pracę pakietów `amsbsy`, `amstext`, `amsthm` i `amscd`.

amsmath pakiet ten stanowi główną część dystrybucji $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a. Definiuje liczne polecenia i otoczenia, wspomagające skład wyrażen i formuł matematycznych. Po jego zadeklarowaniu automatycznie wczytywane są pakiety: `amsopt`, `amsbsy`, `amstext`.

amsopt udostępnia polecenia `\DeclareMathOperator`, `\DeclareMathOperator*`, `\operatorname`, `\operatorname*`, które służą do definiowania własnych operatorów typu logarytmu.

Przegląd pakietów $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a (cd)

- amssymb** udostępnia wiele poleceń tworzących różnego rodzaju symbole matematyczne. Po jego zadeklarowaniu automatycznie wczytywany jest pakiet `amsfonts`.
- amstext** definiuje polecenie `\text`.
- amsthm** zawiera definicję struktur zwanych w $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -u *proclamations*. Jest niezależny od pozostałych pakietów; można go używać nie deklarując innych.
- amsxtra** definiuje znaki diakrytyczne i ozdobniki, umieszczone z boku symboli matematycznych. Ładuje pakiet `amsmath`.

Przegląd pakietów $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a (cd)

Najczęściej używane pakiety to:

- `amsmath` ($\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$)
- `amssymb` ($\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$)

Żeby je zaimplementować w dokumencie należy napisać:

```
\usepackage{amsmath}  
\usepackage{amssymb}
```

Dokumentacja:

Dla `amssymb` plik `amsfndoc.tex` ($\mathcal{A}\mathcal{M}\mathcal{S}\mathcal{F}\mathcal{o}\mathcal{n}\mathcal{t}\mathcal{s}$ User's Guide)

Dla `amsmath` plik `amslldoc.tex` ($\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\mathcal{L}\mathcal{a}\mathcal{t}\mathcal{e}\mathcal{x}$ User's Guide)

Pakiet amsmath

Pakiet amsmath wprowadza wiele usprawnień, których zadaniem jest polepszenie prezentacji matematycznych formuł w dokumencie. W jego skład wchodzi kilka pomocniczych pakietów: amsmath, amstext, amsopn, amsbsy, amscd, oraz amsxtra.

Opcje pakietu amsmath.

- `centertags` – wyrównuje elementy równania do środka w pionie względem całej wysokości równania (domyślna)
- `sumlimits` – umieszcza indeksy górne i dolne symboli takich jak np. suma \sum czy iloczyn \prod nad i pod symbolem (domyślna)
- `nosumlimits` – indeksy z powyższego przykładu umieszczane są obok symboli

Pakiet `amsmath` (cd)

- `intlimits` – granice całkowania powyżej nad i pod symbolem całki
- `nointlimits` – granice obok symbolu całki (domyślna)
- `namelimits` – podobnie jak `sumlimits` ale odnosi się do symboli mających indeksy pod symbolami np. \lim (domyślna)
- `nonamelimits` – analogicznie jak w powyższych przykładach.
- `leqno` – wyrównuje elementy równania do lewej
- `reqno` – wyrównuje elementy równania do prawej
- `fleqn` – równanie zaczyna w nowym akapicie

W dalszym ciągu prezentacji nie zawsze będziemy zaznaczać
która konstrukcja matematyczna jest elementem $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a, a
która $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a

Tryb matematyczny

W TeX-u (również w \LaTeX -u) wyrażenia matematyczne składa się w specjalnym trybie, zw. matematycznym. Ponieważ formuły matematyczne, albo stanowią część akapitu (występują w wierszu), albo są eksponowane, to wyróżniamy dwa rodzaje tego trybu:

- 1 **wewnątrzakapitowy (tekstowy)** – dla symboli podstawowej wielkości w wierszu tekstu;
- 2 **eksponowany** – dla symboli podstawowej wielkości w osobnym wierszu.

Tryb wewnątrzakapitowy

W trybie wewnątrzakapitowym wyrażenia są składane w bieżącym akapicie. W \LaTeX -u wyrażenia matematyczne w trybie wewnątrzakapitowym składa się w środowisku **math**, tj. zaczynamy poleceniem `\begin{math}`, a kończymy poleceniem `\end{math}`. Dla wygody, można również używać skrótu `\(... \)`. Najpopularniejsze jest używanie polecenia \TeX -a, tj. tryb wewnątrzakapitowy zaczynać i kończyć znakiem $\$$.

Tryb wewnątrzakapitowy (cd)

Pisząc:

Stosując rozwinięcie

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

otrzymujemy \ldots

otrzymamy:

Stosując rozwinięcie $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$ otrzymujemy \ldots

Jak widać, formuła jest umieszczona wewnątrz bieżącego wiersza.

Tryb eksponowany

W trybie eksponowanym formuły wstawiane są w osobnym wierszu. Tryb ten występuje w wielu specjalnych środowiskach. Najprostszym środowiskiem (stosowanym do formuł mieszczących się w jednym wierszu) jest środowisko **displaymath**, tj. formułę matematyczną umieszczamy pomiędzy poleceniami `\begin{displaymath}`, a `\end{displaymath}`. Dla wygody, można również używać skrótu, `\[... \]`. Równoważne mu jest środowisko $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -a: **equation***. Nie należy natomiast używać polecenia $\mathcal{T}\mathcal{E}\mathcal{X}$ -a, `$$... $$`, gdyż jest ono niezgodne z $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ -em.

Tryb eksponowany (cd)

Podobnie jak poprzednio, polecenie

```
\begin{displaymath}  
e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}  
\end{displaymath}
```

lub

```
\[  
e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}  
\]
```

Tryb eksponowany

zostanie skompilowane jako

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

Formuła jest w tym przypadku umieszczona w oddzielnym wierszu. Zauważmy, że zmienił się nie tylko sposób osadzenia formuły, ale także wielkość i umiejscowienie granic sumowania oraz wielkość czcionki w liczniku i mianowniku ułamka.

Style trybu matematycznego

W \TeX -u istnieje osiem różnych sposobów składu matematycznego. Są one nazywane *stylami* oraz są normalnymi i zacieśnionymi wersjami stylu eksponowanego, tekstowego, indeksowego i podwójnego indeksu. Wersji zacieśnionej używa się do mianowników ułamków, wyrażeń pod pierwiastkami i w podobnych sytuacjach. \TeX nie podnosi w wersji zacieśnionej indeksów zbyt wysoko.

Do składania indeksów pierwszego rzędu \LaTeX używa stylu indeksowego, natomiast do składania indeksów wyższych rzędów – stylu podwójnego indeksu.

Style trybu matematycznego (cd)

Jawne użycie jednego ze stylów trybu matematycznego wymuszają następujące polecenia:

- Styl eksponowany – **$\text{\texttt{\displaystyle}}$**
- Styl tekstowy – **$\text{\texttt{\textstyle}}$**
- Styl indeksowy – **$\text{\texttt{\scriptstyle}}$**
- Styl podwójnego indeksu – **$\text{\texttt{\scriptscriptstyle}}$**

Kroje czcionek

W trybie matematycznym \LaTeX składa tekst kursywą matematyczną. Różni się ona od kursywy stosowanej w zwykłym tekście tym, że litery są nieco większe, a odstępy wyznaczone między nimi są inaczej.

Np. słowo „*toffi*” w trybie matematycznym wygląda inaczej niż złożone zwykłą kursywą.

Takie zachowanie \LaTeX -a spowodowane jest tym, że w trybie matematycznym pojedyncze litery najczęściej oznaczają zmienne, a ich następstwo oznacza operację mnożenia.

Kroje czcionek(cd)

\LaTeX w trybie matematycznym może używać również następujących krojów czcionek:

prosty	<code>\mathrm{ABC}</code>	ABC
półgruby	<code>\mathbf{ABC}</code>	ABC
bezseryfowy	<code>\mathsf{ABC}</code>	ABC
maszynowy	<code>\mathtt{ABC}</code>	ABC

Kursywę matematyczną włącza polecenie `\mathsl`. Wszystkie te polecenia działają na pojedynczym argumencie.

Kroje czcionek (cd)

Podstawowy \LaTeX zawiera polecenie `\mathcal` dla pisma kaligraficznego. Pakiet **eucal** redefiniuje to polecenie by korzystało z czcionki Euler Script. Natomiast pakiet **eucal** z opcją **mathsc**, nie redefiniuje polecenia `\mathcal`, a czcionki Euler Script są dostępne przez polecenie `\mathsc`. Litery „tablicowe” otrzymujemy po załadowaniu pakietu **amsfonts**, za pomocą polecenia `\mathbb`. Natomiast litery gotyckie uzyskujemy za pomocą polecenia `\mathfrak`, po załadowaniu pakietu **amsfonts** lub pakietu **eufrak**.

Kroje czcionek (cd)

Argumenty polecenia `\mathfrak` mogą zawierać małe i duże litery, a pozostałych poleceń tylko duże litery.

tablicowe	<code>\mathbb{ABC}</code>	\mathbb{ABC}
kaligraficzne (Euler)	<code>\mathscr{ABC}</code>	\mathscr{ABC}
kaligraficzne	<code>\mathcal{ABC}</code>	\mathcal{ABC}
gotyckie	<code>\mathfrak{ABC\ abc}</code>	$\mathfrak{ABC\ abc}$

Odstępy poziome w trybie matematycznym

W trybie matematycznym \LaTeX ignoruje wpisane spacje i formatuje wzór w sposób, który uważa za najlepszy. Można go korygować za następujących pięciu poleceń. Uzyskujemy przy ich pomocy odstęp poziomy takiej wielkości jak pokazany między poziomymi kreskami:

$ \backslash,$ wąski	$ \backslash:$ średni
$ \backslash!$ wąski ujemny (wstecz)	$ \backslash;$ szeroki

Dodatkową spację (podobnie jak w trybie akapitowym) uzyskujemy za pomocą symbolu: \backslash i następującej po nim spacji. Odstęp $\backslash,$ można również używać w trybie akapitowym.

Odstępy poziome w trybie matematycznym (cd)

Polecenia `\quad` i `\qquad` dają większe odstępy (można używać ich również w trybie akapitowym). Polecenie `\quad` tworzy odstęp równy wielkości bieżącego pisma, np. w piśmie 12 pt daje 12 pt odstępu. Polecenie `\qquad` tworzy odstęp dwukrotnie większy.

Pisząc

`|x|=x \quad x \geq 0`

otrzymujemy

$$|x| = x \quad x \geq 0$$

Ręczne łamanie wiersza w trybie wewnętrzakapitowym

Jeżeli chcemy przejść do następnego wiersza w trybie
wewnętrzakapitowym należy zastosować polecenie `\\`:

Potęgi i indeksy

Potęgi oraz indeksy są tym samym co *frakcja górna* oraz *frakcja dolna* w trybie tekstowym. Aby umieścić coś do górnego indeksu, należy użyć symbolu \wedge , np. pisząc $\mathbf{a}^{\mathbf{m}}$ otrzymujemy a^m . Indeks dolny tworzymy przy pomocy symbolu $_$, np. pisząc $\mathbf{a}_{\mathbf{n}}$ otrzymujemy a_n . Istnieje możliwość połączenia indeksu górnego z indeksem dolnym. Kolejność indeksów nie odgrywa roli. Jeśli jakieś wyrażenie ma się znaleźć w dolnym bądź górnym indeksie, należy je umieścić w klamrach $\{ \}$.

Potęgi i indeksy(cd)

Pisząc

$$a_{n+1}, x^2, k_{n+1} = n^2 + k_n^2 - k_{n-1}$$

otrzymujemy

$$a_{n+1}, x^2, k_{n+1} = n^2 + k_n^2 - k_{n-1}$$

Obramowania, czyli ramki

Dzięki poleceniu `\boxed` można jakiegokolwiek wyrażenie (np. równanie) w łatwy sposób wyróżnić, umieszczając je w ramce, np. pisząc

```
\[  
\boxed{E = mc^2}  
\]
```

otrzymujemy

$$E = mc^2$$

Ułamki

Ułamek tworzymy używając komendy
`\frac{licznik}{mianownik}`. Pisząc

`\frac{2}{4}`

otrzymujemy

$$\frac{2}{4}.$$

Ułamki (cd)

Można zagnieżdżać ułamki w ułamkach: pisząc

$$\backslash\text{frac}\{\backslash\text{frac}\{1\}{x}+\backslash\text{frac}\{1\}{y}\}{y-z}$$

otrzymujemy

$$\frac{\frac{1}{x} + \frac{1}{y}}{y - z}$$

oraz jako indeksy górne i dolne, itp.: pisząc

$$y^{\{\backslash\text{frac}\{1\}{2}\}}$$

otrzymujemy

$$y^{\frac{1}{2}}$$

Ułamki (cd)

Do utworzenia prostego ułamku umieszczonego w tekście (niematematycznym), można użyć potęg i indeksów., np. pisząc $\{\}^{\mathbf{2}}/\{\}_{-\mathbf{13}}$ otrzymujemy $^2/_{13}$.

Ułamki (cd)

Pakiet `amsmath` zawiera wygodne skróty do komend tworzących ułamki:

`\tfrac` dla `\textstyle\frac{\dots}{\dots}`,
`\dfrac` dla `\displaystyle\frac{\dots}{\dots}`.

Ułamki (cd)

Pisząc

`\sqrt{\dfrac{1}{x}\cos t}\quad \sqrt{\frac{1}{x}\cos t}`

otrzymujemy $\sqrt{\frac{1}{x}\cos t}$ $\sqrt{\frac{1}{x}\cos t}$

Podobnie pisząc

`\[\sqrt{\tfrac{1}{x}\cos t}\quad \sqrt{\frac{1}{x}\cos t}\]`

otrzymujemy

$$\sqrt{\frac{1}{x}\cos t} \quad \sqrt{\frac{1}{x}\cos t}$$

Ułamki łańcuchowe

Ułamki łańcuchowe zapisujemy używając komendy `\cfrac`:
pisząc

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + a_4}}}$$

otrzymujemy

$$x = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \frac{1}{a_3 + a_4}}}$$

Symbol Newtona

Symbol Newtona tworzony jest przy pomocy polecenia
`\binom`: pisząc

$$\backslash\mathrm{binom}\{n\}\{k\}=\backslash\mathrm{frac}\{n!\}\{k!(n-k)!\}$$

otrzymujemy

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Symbol Newtona (cd)

Pakiet `amsmath` zawiera wygodne skróty do komend tworzących symbol Newtona:

`\tbinom` dla `\textstyle\binom{...}{...}`,
`\dbinom` dla `\displaystyle\binom{...}{...}`.

Symbol Newtona (cd)

Pisząc

`\dbinom{k}{n+1}\quad \binom{k}{n+1}`

otrzymujemy

$$\binom{k}{n+1} \quad \binom{k}{n+1}$$

Podobnie pisząc

`\[\tbinom{k}{n+1}\quad \binom{k}{n+1} \]`

otrzymujemy

$$\binom{k}{n+1} \quad \binom{k}{n+1}$$

Struktury „ułamkopodobne”

Poleceniem `\genfrac` można tworzyć dowolne struktury „ułamkopodobne”. Ma ono następującą składnię:

`\genfrac{ldelim}{rdelim}{grubość}{styl}{licznik}{mianownik}`

Pisząc

`\genfrac{[]{}{2pt}{1}{1}{n+1}`

otrzymujemy

$$\left[\frac{1}{n+1} \right]$$

Pierwiastki

Polecenie `\sqrt[st_pierwiastka]{}` tworzy pierwiastek otaczający dane wyrażenie. Jeżeli stopień pierwiastka nie jest podany, to utworzony zostanie pierwiastek kwadratowy. Pisząc

`\sqrt{\frac{a}{b}}`, `\sqrt[n]{1+x+x^2+x^3+\ldots}`, `\sqrt[3]{x+y}`,

otrzymamy

$$\sqrt{\frac{a}{b}}, \quad \sqrt[n]{1+x+x^2+x^3+\ldots}, \quad \sqrt[3]{x+y},$$

Pierwiastki (cd)

Ponieważ komenda `\sqrt` nie zawsze daje pożądane efekty, w $\mathcal{A}\mathcal{M}\mathcal{S}$ - $\mathcal{L}\mathcal{A}\mathcal{T}\mathcal{E}\mathcal{X}$ rozszerzono ją o polecenia `\leftroot` i `\uproot` pozwalające na pozycjonowanie stopnia pierwiastka.

Pisząc

```
[ \sqrt[\varphi]{x} \quad \quad \quad \sqrt[\leftroot{2}\uproot{4}]{\varphi}{x} ]
```

otrzymujemy

$$\sqrt[\varphi]{x} \qquad \sqrt[\leftroot{2}\uproot{4}]{\varphi}{x}$$

Operatory binarne

Działania dodawania jak i odejmowania są definiowane w naturalny sposób, np.: pisząc $\mathbf{a+b-c=0}$ otrzymujemy $a + b - c = 0$.

Pozostałe operatory:

\cdot	<code>\cdot</code>	\circ	<code>\circ</code>	\cap	<code>\cap</code>	\wedge	<code>\wedge</code>
\times	<code>\times</code>	\bullet	<code>\bullet</code>	\cup	<code>\cup</code>	\oplus	<code>\oplus</code>
\star	<code>\star</code>	\div	<code>\div</code>	\vee	<code>\vee</code>	\otimes	<code>\otimes</code>
\times	<code>\times</code>	$*$	<code>\ast</code>	\ominus	<code>\ominus</code>	\odot	<code>\odot</code>
\diamond	<code>\diamond</code>						

Można również utworzyć operator $\underline{\vee}$ pisząc `\underline{\vee}`.

Operatory binarne

Przykłady:

Kod	Rezultat
$x \cdot y = 0$	$x \cdot y = 0$
$A \cap B = B \cap A$	$A \cap B = B \cap A$

Pisząc $A \cup B = \{x : (x \in A) \vee (x \in B)\}$
otrzymujemy

$$A \cup B = \{x : (x \in A) \vee (x \in B)\}$$

Relacje

$<$	<code><</code>	$>$	<code>></code>	$=$	<code>=</code>
\leq	<code>\leq</code> lub <code>\le</code>	\geq	<code>\ge</code> lub <code>\geq</code>	\equiv	<code>\equiv</code>
\ll	<code>\ll</code>	\gg	<code>\gg</code>	\doteq	<code>\doteq</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>
\subset	<code>\subset</code>	\supset	<code>\supset</code>	\approx	<code>\approx</code>
\subseteq	<code>\subseteq</code>	\supseteq	<code>\supseteq</code>	\cong	<code>\cong</code>
\sqsubset	<code>\sqsubset</code>	\sqsupset	<code>\sqsupset</code>	\Join ¹	<code>\Join</code> ¹

¹Dostępne do załadowaniu pakietu `latexsym`.

Relacje (cd)

\sqsubseteq	<code>\sqsubseteq</code>	\sqsupseteq	<code>\sqsupseteq</code>	\bowtie	<code>\bowtie</code>
\in	<code>\in</code>	\ni	<code>\ni</code>	\propto	<code>\propto</code>
\vdash	<code>\vdash</code>	\dashv	<code>\dashv</code>	\models	<code>\models</code>
$ $	<code>\mid</code>	\parallel	<code>\parallel</code>	\perp	<code>\perp</code>
$:$	<code>:</code>	\notin	<code>\notin</code>	\neq	<code>\neq</code> lub <code>\ne</code>
\leqslant	<code>\leqslant</code>	\geqslant	<code>\geqslant</code>		

Odpowiednie symbole negacji relacji można utworzyć poprzedzając powyższe symbole polecenie `\not`.

Relacje (cd)

Przykłady:

Kod	Wynik
<code>x\parallel y</code>	$x \parallel y$
<code>A\sim B</code>	$A \sim B$

Przystawanie modulo

Kod

```
a\equiv b \pmod{11}
```

```
a\equiv_{11} b
```

Wynik

$$a \equiv b \pmod{11}$$
$$a \equiv_{11} b$$

Strzałki

\leftarrow	<code>\leftarrow</code>	\leftarrow	<code>\gets</code>
\rightarrow	<code>\rightarrow</code>	\rightarrow	<code>\to</code>
\Leftrightarrow	<code>\Leftarrow</code>	\Rightarrow	<code>\Rightarrow</code>
\longleftarrow	<code>\longleftarrow</code>	\longrightarrow	<code>\longrightarrow</code>
\Longrightarrow	<code>\Longrightarrow</code>	\Longleftarrow	<code>\Longleftarrow</code>
\leftrightarrow	<code>\leftrightarrow</code>	\longleftrightarrow	<code>\longleftrightarrow</code>
\Leftrightarrow	<code>\Longleftrightarrow</code>	\Leftrightarrow	<code>\iff</code>
\mapsto	<code>\mapsto</code>	\leftharpoonup	<code>\leftharpoonup</code>
\longmapsto	<code>\longmapsto</code>	\nearrow	<code>\nearrow</code>
\swarrow	<code>\swarrow</code>	\searrow	<code>\searrow</code>
\nwarrow	<code>\nwarrow</code>	\uparrow	<code>\uparrow</code>
\downarrow	<code>\downarrow</code>		

Napis nad strzałką

Aby umieścić napis nad strzałką stosujemy poniższe komendy:

Kod

$f(x) \xrightarrow{T_{\{[1,5]\}}} g(x)$
 $X \xleftarrow{temp} Y$

Wynik

$f(x) \xrightarrow{T_{[1,5]}} g(x)$
 $X \xleftarrow{temp} Y$

Wektory oraz linie nad/pod wyrażeniami

W celu uzyskania strzałki nad zmienną stosujemy polecenie

`\vec{}`, np. pisząc `\vec{a}` otrzymujemy \vec{a} .

Jeżeli chcemy uzyskać kreskę pod lub nad wyrażeniem wystarczy odpowiednio zastosować polecenie `\underline` lub `\overline`.

Np. pisząc:

`\underline{abc}`, `\overline{cde}`,
`\overline{\overline{\Omega}}`

otrzymujemy

$$\underline{abc}, \overline{cde}, \overline{\overline{\Omega}}$$

Wektory oraz linie nad/pod wyrażeniami(cd)

Można również stosować nawiasy klamrowe pod/nad wyrażeniem, np pisząc:

$$n \cdot a = \underbrace{a + a + \dots + a}_{n},$$

$$a^m = \overbrace{a \cdot a \cdot \dots \cdot a}^m$$

otrzymujemy

$$n \cdot a = \underbrace{a + a + \dots + a}_n, \quad a^m = \overbrace{a \cdot a \cdot \dots \cdot a}^m$$

Wektory oraz linie nad/pod wyrażeniami(cd)

W celu umieszczenia wyrażenia nad innym wyrażeniem stosujemy polecenie `\stackrel{}{}`, np. : zaznaczenie (nad znakiem równości) użycia reguły de l'Hospitala w obliczeniach:

$$\stackrel{[H]}{=}$$

uzyskujemy stosujemy kod:

$$\stackrel{[H]}{=}$$

Do wstawienia wyrażenia poniżej innego służy komenda `\mathop{}_{}`, np.pisząc:

$$\frac{1}{n+1} \mathop{\longrightarrow}_{n \rightarrow \infty} 0$$

otrzymujemy

$$\frac{1}{n+1} \mathop{\longrightarrow}_{n \rightarrow \infty} 0$$

Różne znaki

\forall	<code>\forall</code>	\hbar	<code>\hbar</code>	\emptyset	<code>\emptyset</code>	\exists	<code>\exists</code>
∇	<code>\nabla</code>	∂	<code>\partial</code>	∞	<code>\infty</code>	\int	<code>\int</code>
\oint	<code>\oint</code>	\sum	<code>\sum</code>	\prod	<code>\prod</code>	\aleph	<code>\aleph</code>
\Re	<code>\Re</code>	\Im	<code>\Im</code>	ℓ	<code>\ell</code>	\angle	<code>\angle</code>
\neg	<code>\neg</code>	\dots	<code>\ldots</code>	\ddots	<code>\ddots</code>	\vdots	<code>\vdots</code>
$'$	<code>\prime</code>	$\%$	<code>\%</code>	\nmid	<code>\nmid</code>		

Różne znaki(cd)

Przykłady:

Pisząc

$$\forall x \in \mathbb{R} \quad x^2 \geq 0$$

otrzymujemy

$$\forall x \in \mathbb{R} \quad x^2 \geq 0$$

Podobnie pisząc

$$\text{Jeżeli } z = a + bi, \text{ to } \operatorname{Re}(z) = a, \operatorname{Im}(z) = b$$

, otrzymujemy

$$\text{Jeżeli } z = a + bi, \text{ to } \Re(z) = a, \Im(z) = b$$

Granice

Aby uzyskać symbol granicy: $\lim_{x \rightarrow \infty} f(x)$ należy wpisać kod:
`\lim_{x\to\infty}f(x)`, np. pisząc

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$$

otrzymujemy

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e.$$

Sumy, iloczyny i inne „duże” operatory

Komendy `\sum` oraz `\prod` wstawiają odpowiednio symbole sumy oraz iloczynu z granicami zdefiniowanymi, odpowiednio, przy pomocy symboli `^` oraz `_`, pisząc:

$$\sum_{i=1}^{10} t_i$$

otrzymujemy

$$\sum_{i=1}^{10} t_i$$

Sumy, iloczyny i inne „duże” operatory (cd)

Pisząc

$\sum_{j=1}^n \frac{\partial f}{\partial x_j} \frac{\partial}{\partial t_i}$, $i=1,2,\dots,k$

otrzymujemy

$$\sum_{j=1}^n \frac{\partial f}{\partial x_j} \frac{\partial}{\partial t_i}, \quad i = 1, 2, \dots, k$$

Podobnie pisząc

$$\prod_{n=0}^k n = 0$$

otrzymujemy

$$\prod_{n=0}^k n = 0$$

Sumy, iloczyny i inne „duże” operatory (cd)

Istnieją jeszcze inne „duże” operatory, których używa się w podobny sposób:

\sum	<code>\sum</code>	\prod	<code>\prod</code>	\coprod	<code>\coprod</code>
\bigoplus	<code>\bigoplus</code>	\bigotimes	<code>\bigotimes</code>	\bigodot	<code>\bigodot</code>
\bigcup	<code>\bigcup</code>	\bigcap	<code>\bigcap</code>	\biguplus	<code>\biguplus</code>
\bigsqcup	<code>\bigsqcup</code>	\bigvee	<code>\bigvee</code>	\bigwedge	<code>\bigwedge</code>

Sumy, iloczyny i inne „duże” operatory (cd)

Polecenie `\substack` pozwala używać `\\`, aby ustalać wielolinijkowe granice, np pisząc:

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} P(i,j)$$

otrzymujemy

$$\sum_{\substack{0 < i < m \\ 0 < j < n}} P(i,j)$$

Sumy, iloczyny i inne „duże” operatory (cd)

Suma mnogościowa: $\bigcup_{s=1}^n A_s = A_1 \cup A_2 \cup \dots \cup A_n$
otrzymujemy pisząc

`\bigcup_{s=1}^n A_s = A_{\{1\}} \cup A_{\{2\}} \cup \ldots`
`\cup A_{\{n\}}`

Iloczyn mnogościowy (przekrój): $\bigcap_{s=1}^n A_s$ otrzymujemy pisząc

`\bigcap_{s=1}^n A_s`

Chcąc uzyskać tylko symbole \sum , \prod , \cap , \cup wystarczy wpisać odpowiednio:

`\sum`, `\prod`, `\bigcap`, `\bigcup`

Sumy, iloczyny i inne „duże” operatory (cd)

Operatory `\bigvee` oraz `\bigwedge` mogą być używane jako kwantyfikatory \vee oraz \wedge , np. pisząc:

$$\bigwedge_{x \in \mathbb{R}} e^x > 0$$

otrzymujemy

$$\bigwedge_{x \in \mathbb{R}} e^x > 0$$

Gdy chcemy umieścić wzór $\bigwedge_{x \in \mathbb{R}} e^x > 0$ w linii tekstu należy napisać kod: `\$ \bigwedge \limits_{x \in \mathbb{R}} e^x > 0 \$`

Całki

Polecenie `\int` wstawia symbol całki z granicami zdefiniowanymi, odpowiednio, przy pomocy symboli `^` oraz `_`, np. pisząc:

$$\int_0^\infty f(x) dx$$

otrzymujemy

$$\int_0^\infty f(x) dx$$

Aby granice całkowania były zapisane pod oraz nad symbolem (tak jak np. w sumie), należy użyć polecenia `\limits`, pisząc:

$$\int\limits_{\{a\}}^{\{b\}} f(x) dx$$

otrzymujemy

$$\int_a^b f(x) dx$$

Całki (cd)

Istnieją również inne symbole całki:

$$\begin{array}{llll} \int & \backslash\text{int} & \oint & \backslash\text{ooint} \\ \iiint & \backslash\text{iiint} & \iiint & \backslash\text{iiiint} \\ \int \cdots \int & \backslash\text{idotsint} \end{array}$$

Pisząc

$$\begin{array}{l} \backslash\text{iiint}_{\{D\}} g(x,y) \backslash, \text{d}x\text{d}y, \backslash\text{quad} \\ \backslash\text{int}\limits_{\{0\}^{\{1\}}}\backslash\text{int}\limits_{\{2\}^{\{4\}}} g(x,y) \backslash, \text{d}x\text{d}y \end{array}$$

otrzymujemy

$$\iint_D g(x,y) \text{d}x\text{d}y, \quad \int_0^1 \int_2^4 g(x,y) \text{d}x\text{d}y$$

Nawiasy i ograniczniki

We wzorach matematycznych najczęściej używane są nawiasy okrągłe, kwadratowe oraz klamrowe. Np. pisząc

$$(a+b)^2=a^2+2ab+b^2, \ x \in [a,b], \ A=\{x \in R: \ x>0\}$$

otrzymujemy

$$(a+b)^2 = a^2 + 2ab + b^2, \ x \in [a, b], \ A = \{x \in R : \ x > 0\}$$

Nawiasy i ograniczniki (cd)

W składzie komputerowym \LaTeX można używać również innych nawiasów:

(())	[[
]]	{	\{ lub \lbrace	}	\} lub \rbrace
[\lfloor]	\rfloor	<	\langle
>	\rangle	⌈	\lceil	⌋	\rceil
			\		

Zauważmy, że nawiasy | i || mają taką samą wersję prawą i lewą.

Skalowanie nawiasów

Ponieważ wzory matematyczne są często różnych rozmiarów, to nawiasy powinny również przyjmować różne rozmiary. Efekt automatycznego dopasowania rozmiaru nawiasów do wyrażenia matematycznego które otaczają można osiągnąć używając poleceń `\left` i `\right` w połączeniu z dowolnym nawiasem lub ogranicznikiem przedstawionym powyżej, np. pisząc:

```
(\frac{x^{2}}{y^{3}})
\left(\frac{x^{2}}{y^{3}}\right)
```

otrzymujemy

$$\left(\frac{x^2}{y^3}\right),$$

Skalowanie nawiasów (cd)

Podobnie pisząc

$\left\lfloor \frac{1}{x} \right\rfloor$, $\left\lceil -1 \frac{1}{2}; 5 \right\rceil$,
 $\left\{ 1, \frac{1}{2}, \frac{1}{3} \right\}$

otrzymujemy

$$\left\lfloor \frac{1}{x} \right\rfloor, \left\langle -1 \frac{1}{2}, 5 \right\rangle, \left\{ 1, \frac{1}{2}, \frac{1}{3} \right\}$$

a pisząc

$\int_c^d \left[\int_{u(y)}^{v(y)} f(x, y) dx \right] dy$

otrzymujemy

$$\int_c^d \left[\int_{u(y)}^{v(y)} f(x, y) dx \right] dy$$

Skalowanie nawiasów (cd)

Jeśli ma być widoczny tylko nawias po jednej stronie, wtedy niewidoczny nawias określamy kropką (.). Taka sytuacja występuje, gdy wyrażenie jest zbyt duże by zmieścić się w jednej linii, np. pisząc

$\left[\int_c^d \int_{u(y)}^{v(y)} f(x,y) dx dy\right].$

otrzymujemy

$$\left[\int_c^d \int_{u(y)}^{v(y)} f(x,y) dx dy \right.$$

Skalowanie nawiasów (cd)

Można również ręcznie ustawić odpowiednią wielkość nawiasu. W tym celu używa się, odpowiednio, poleceń `\big`, `\Big`, `\bigg`, `\Bigg` z odpowiednim nawiasem, tj. pisząc:

`(\big(\Big(\bigg(\Bigg(\Bigg) \bigg)))`

otrzymujemy

$(((((\Bigg))))))$

Funkcje matematyczne

\sin	$\backslash\sin$	\cos	$\backslash\cos$	tg	$\backslash\operatorname{tg}$
ctg	$\backslash\operatorname{ctg}$	csc	$\backslash\operatorname{csc}$	$\operatorname{arc\,sin}$	$\backslash\operatorname{arcsin}$
$\operatorname{arc\,cos}$	$\backslash\operatorname{arccos}$	$\operatorname{arc\,tg}$	$\backslash\operatorname{arctan}$	\sinh	$\backslash\sinh$
\cosh	$\backslash\cosh$	ctgh	$\backslash\operatorname{coth}$	\sup	$\backslash\sup$
\inf	$\backslash\inf$	\limsup	$\backslash\limsup$	\liminf	$\backslash\liminf$
\log	$\backslash\log$	\lg	$\backslash\lg$	\ln	$\backslash\ln$
\exp	$\backslash\exp$	\det	$\backslash\det$	\deg	$\backslash\deg$
\dim	$\backslash\dim$	hom	$\backslash\operatorname{hom}$	\ker	$\backslash\ker$
\max	$\backslash\max$	\min	$\backslash\min$	\arg	$\backslash\arg$
nwd	$\backslash\operatorname{nwd}$				

Funkcje matematyczne (cd)

Kod	Wynik
<code>\sin 2\pi=0</code>	$\sin 2\pi = 0$
<code>\ln (e)=1</code>	$\ln(e) = 1$

Akcenty matematyczne

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

Małe litery greckie

α	<code>\alpha</code>	θ	<code>\thetaeta</code>	o	<code>o</code>	v	<code>\upsilon</code>
β	<code>\betaeta</code>	ϑ	<code>\varthetaeta</code>	π	<code>\pi</code>	ϕ	<code>\phi</code>
γ	<code>\gamma</code>	ι	<code>\iota</code>	ϖ	<code>\varpi</code>	φ	<code>\varphi</code>
δ	<code>\delta</code>	κ	<code>\kappa</code>	ρ	<code>\rho</code>	χ	<code>\chi</code>
ϵ	<code>\epsilon</code>	λ	<code>\lambda</code>	ϱ	<code>\varrho</code>	ψ	<code>\psi</code>
ε	<code>\varepsilon</code>	μ	<code>\mu</code>	σ	<code>\sigma</code>	ω	<code>\omega</code>
ζ	<code>\zeta</code>	ν	<code>\nu</code>	ς	<code>\varsigma</code>		
η	<code>\eta</code>	ξ	<code>\xi</code>	τ	<code>\tau</code>		

Wielkie litery greckie

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

Polecenia `\newtheorem` i `\newtheorem*`

W tekstach matematycznych występują zwykle twierdzenia, lematy, wnioski, definicje uwagi itp. \LaTeX nie dysponuje oddzielnymi otoczeniami dla każdego z nich, ale udostępnia deklaracje `\newtheorem` i `\newtheorem*` do definiowania przez użytkownika własnych środowisk.

Deklaracje te można umieszczać w dowolnym miejscu dokumentu, ale najlepiej w preambule.

Mają one postać

```
\newtheorem{naz_otocz}[wspol_num]{etykieta}[jeden_podz]  
\newtheorem*{naz_otocz}{etykieta}
```

Polecenia `\newtheorem` i `\newtheorem*` (cd)

Argument `naz_otocz` to nazwa jaką użytkownik nadał otoczeniu, etykieta jest tekstem którym opatruje się to otoczenie. W przypadku polecenia `\newtheorem`, etykieta jest licznikiem, czyli kolejne etykiety zaopatrywane są numerem, np. Tw. 1, Tw. 2 itd. Opcjonalny argument `jedn_pod` sprawia, że otoczenia danego rodzaju są numerowane w obrębie określonej jednostki podziału tekstu, np. chapter, section. Brak tego argumentu powoduje, że stosowana jest ciągła numeracja w całej pracy. Opcjonalny argument `wspol_num` umożliwia wspólną numerację różnych otoczeń, np. typu twierdzeń.

Argumenty opcjonalne `jedn_podz` i `wspol_num` nie mogą występować jednocześnie.

Polecenia `\newtheorem` i `\newtheorem*` (cd)

Np. umieszczenie w preambule

```
\newtheorem{thm}{TWIERDZENIE}[chapter]  
\newtheorem{lem}[thm]{LEMAT}  
\newtheorem{cor}[thm]{WNIOSEK}
```

określa trzy środowiska: `thm`, `lem` i `cor`, które będą wspólne numerowanie, oddzielnie w każdym rozdziale. Mają one etykiety, odpowiednio: TWIERDZENIE, LEMAT i WNIOSEK.

Style otoczeń tworzonych przy pomocy poleceń `\newtheorem` i `\newtheorem*`

Wszystkie otoczenia zdefiniowane przez polecenia
`\newtheorem` i `\newtheorem*` mają podobny wygląd. Pakiet
`amsthm` umożliwia za pomocą polecenia

$$\text{\texttt{\textbackslash theoremstyle\{naz_stylu\}}}$$

określenie stylu dalej określonych otoczeń.

Style otoczeń tworzonych przy pomocy poleceń `\newtheorem` i `\newtheorem*` (cd)

Pre-definiowane są 3 style

- `plain` zalecany do twierdzeń, lematów, wniosków itp.
(domyślny);
- `definition` zalecany do definicji i przykładów;
- `remark` zalecany do uwag i notatek (notes).

Style otoczeń tworzonych przy pomocy poleceń `\newtheorem` i `\newtheorem*`

Np. umieszczenie w preambule

```
\theoremstyle{definition}  
\newtheorem{defn}{DEFINICJA}[chapter]  
\newtheorem{exmp}{Przykład}[chapter]
```

określa dwa środowiska `defn` i `exmp`, które będą składane w stylu `definition`. Mają one etykiety: `DEFINICJA` i `Przykład` i są numerowane oddzielnie w obrębie każdego rozdziału.

Środowisko proof

Pakiet **amsthm** udostępnia specjalne środowisko do umieszczania dowodów. Ma ono postać:

```
\begin{proof}
```

Tutaj zamieszczamy dowód.

```
\end{proof}
```

Środowisko to rozpoczyna się złożonym italiikiem słowem *Proof*, a kończy białym kwadratem (Q.E.D Symbol). Jeżeli używany jest pakiet `babel`, to słowo *Proof*, zostanie zastąpione odpowiednikiem we właściwym języku, np. słowem *Dowód*.

Środowisko proof (cd)

Możemy również sami nazwać dowód. Np. pisząc

```
\begin{proof}[Dowodzik]
```

Tutaj zamieszczamy dowód.

```
\end{proof}
```

zastąpimy słowo *Proof* lub *Dowód* (gdy używamy pakietu `babel`) słowem *Dowodzik*.

Środowisko proof (cd)

Gdy używamy pakietu **polski**, to aby otrzymać na początku słowo *Dowód* piszemy

```
\begin{proof}[Dowód]
```

Tutaj zamieszczamy dowód.

```
\end{proof}
```