

L^AT_EX na kolorowo

Włodzimierz Macewicz

Streszczenie

Artykuł przedstawia możliwości zastosowania kolorów w dokumencie przetwarzanym L^AT_EXem. Za pomocą licznych przykładów starałem się zilustrować kolorowanie różnych elementów.

Kolorowanie umożliwiając makroinstrukcje zdefiniowane w pakiecie `color` i `colortbl`. Oprócz wymienionych pakietów są jeszcze: `colordvi` i `colortab`, ale nie wnoszą one żadnych dodatkowych możliwości.

Nie czuję się na siłach odpowiedzieć na pytanie „po co?” lub „dlaczego?” kolorować, skoncentruję się na udzieleniu odpowiedzi na pytanie „jak?”

1. Wprowadzenie

Nie jest moim zamierzeniem narzucanie (ani nawet sugerowanie) sposobu wykorzystania opisywanych mechanizmów do kolorowania. W poniższym opracowaniu starałem się zilustrować przykłady, używając kontrastowych kolorów, przez co jest on nieco pstrokaty – w żadnym wypadku proszę nie traktować go jako wzór (zgodnie z powiedzeniem „róbcie tak jak ja mówię, a nie tak jak ja robię” ; -}).

O ile kolorowanie pisma jest stosunkowo proste i polega na wstawieniu wywołania makroinstrukcji `color` w odpowiednie miejsce, o tyle kolorowanie tła jest kłopotliwe; za pomocą istniejących pakietów można zrealizować ten efekt tylko dla elementów które dają się zamknąć w „pudełko”¹ (oprócz poniżej opisanych pakietów można wykorzystać konstrukcje pakietów `shadethm` i `fancybox`), nie widzę możliwości kolorowania tła na obszarze ciągłym przechodzącym ze strony na stronę (może ktoś zmodyfikuje konstrukcje pakietu `changebar`). Pewną protezę można zrealizować za pomocą konstrukcji `longtable`.

Wprowadzenie poleceń kolorujących standardowe konstrukcje L^AT_EXa wymaga modyfikacji kodu, co prawda w kilku miejscach umieszczono wywołania makroinstrukcji, które po zdefiniowaniu mo-

¹ Odpowiednie zdefiniowanie położenia i grubości linii prezentowanej standardowo na marginesie, tworzonej za pomocą poleceń pakietu `changebar` pozwala na uzyskanie tego efektu, ale niestety kolor jest nakładany „na” dany obszar, a nie „pod” – efekt jest taki że sam tekst staje się niewidoczny. Wydaje mi się, że użycie analogicznego mechanizmu – nieco zmodyfikowanego – pozwoliłoby na uzyskanie kolorowania tła.

głyby służyć do zmiany koloru, ale ich lokalizację trudno określić inaczej niż przypadkową.

Wszelkie przedstawione modyfikacje mają sens nakładek, czyli po oryginalnej definicji (wywołaniu pakietu) należy wprowadzić przedstawiony kod. Proszę pamiętać o tym, aby **nie modyfikować oryginalnego kodu pakietów**.

2. Pakiet `color`

Pakiet definiuje kilka makroinstrukcji umożliwiających kolorowanie pisma lub tła oraz definiowanie własnych kolorów. W przypadku pdfL^AT_EXa proces kolorowania odbywa się jednocześnie ze składem, natomiast w przypadku T_EXa jest realizowany przez postprocesor przetwarzający plik `dvi` (np. program `dvips`) co w złośliwych wypadkach, może dawać **różne** wyniki końcowe.

2.1. Opcje pakietu

Konstrukcje pakietu `color` realizują zakładane efekty, wstawiając do pliku wynikowego makroinstrukcje `\special` z odpowiednimi parametrami (dla pdfL^AT_EX-a lub `dvips`-a). Dla systemu pdfL^AT_EX i T_EX + `dvips` zostało to zestandaryzowane (zarówno pdfL^AT_EX jak i `dvips` są dostępne we wszystkich systemach operacyjnych). W przypadku innych programów przetwarzających należy wstawić odmienne parametry, o czym powinien zostać poinformowany pakiet `color`, aby do pliku wynikowego `dvi` został wstawiony odpowiedni kod.

Opcje związane z implementacją systemu

Zbiór `dvi` będzie przetwarzany za pomocą programu:

`dvips` – `dvips` (włącza opcję `dvipsnames`),
`xdvi` – `xdvi` (włącza opcję `monochrome`),
`dvipdf` – `dvipdf` autorstwa Sergieja Lesenka,
`dvipdfm` – `dvipdf` autorstwa Marka Wick’a,
`pdftex` – pdfL^AT_EX,
`dvipsone` – Y&Y T_EX,
`dviwindo` – jw,
`emtex` – emL^AT_EX (już historia),
`dviwin` – jw,
`oztex` – ozL^AT_EX na Macintoshu,
`textures` – Blue sky’s Textures na Macintoshu,
`pctexps` – PPT na PC,
`pctexwin` – jw,
`pctexhp` – jw,
`pctex32` – jw,
`truetex` – Kinch T_EX na PC,

tcidvi - jw,
vtex - VTeX.

W większości przypadków działanie opcji specyficznych dla różnych implementacji jest mi nieznane - ich listę wzięłem z dokumentacji pakietu [4].

Pozostałe opcje sterujące

debugshow - włącza śledzenie przetwarzania,

monochrome - dezaktywuje wszelkie polecenia zmieniające kolor,

usenames - powoduje że wszystkie nazwy kolorów definiowanych jako named będą dostępne bezpośrednio (tak jak kolory definiowane explicit - poleceniem `\definecolor`); w przeciwnym razie należy poleceń zmieniających kolor z opcją named np. `\color[named]{kolor}`,

dvipsnames - dołącza zbiór definiujący kolory („named colors”) zgodne z kolorami definiowanymi przez program przetwarzający (dvips w tym wypadku),

nodvipsnames - kasuje definicje kolorów dostępnych jako „named colors”

2.2. Polecenia zmiany koloru

Kolory można definiować na dwa sposoby: za pomocą poleceń `\definecolor` lub `\DefineNamedColor`. Drugi ze sposobów jest praktycznie niedostępny użytkownikowi (jest on związany ze szczegółami implementacyjnymi); jednak w ten sposób zostało zdefiniowanych wiele kolorów, szkoda więc z nich rezygnować, w celu ich użycia należy wywoływać polecenia zmiany koloru z opcjonalnym parametrem `model` o wartości `named`² (np.: `\color[named]{Salmon}`) lub użyć opcji `usenames` przy deklaracji wywołania pakietu. Dla porządku podając listę poleceń uwzględniam obie możliwości:

`\color{nazwa koloru}`

`\color[model]{nazwa koloru}` polecenie powoduje zmianę koloru pisma, działa analogicznie jak polecenia zmiany kroju np. `\bfseries`;

Przykład:

`{\color{red} tekst w kolorze czerwonym}`

tekst w kolorze czerwonym

`{\color[named]{Green}`

tekst w kolorze zielonym}

tekst w kolorze zielonym

² można zdefiniować inne pseudomodele, ale wtedy dokument staje się nieprzenośny między różnymi implementacjami systemu TeX.

`\textcolor{nazwa koloru}{tekst}`

`\textcolor[model]{nazwa koloru}{tekst}`

polecenie powoduje zmianę koloru pisma, działa analogicznie jak polecenia zmiany kroju np. `\textbf{tekst}`;

Przykład:

`\textcolor{red}`

{tekst w kolorze czerwonym}

tekst w kolorze czerwonym

`\textcolor[named]{Green}`

{tekst w kolorze zielonym}

tekst w kolorze zielonym

`\colorbox{kolor tła}{tekst}`

`\colorbox[model]{kolor tła}{tekst}`

polecenie powoduje zmianę koloru tła „pudełka” zawierającego *tekst* (będzie miało ono długość nominalną);

Przykład:

`\colorbox{red}{tło w kolorze czerwonym}`

tło w kolorze czerwonym

`\colorbox[named]{Yellow}`

{tło w kolorze żółtym}

tło w kolorze żółtym

pole tła jest większe od obszaru tekstu o wartość parametru `\fboxsep`. Parametr `tekst` może być bardziej złożoną konstrukcją np. `\minipage`, dzięki temu możemy zdefiniować większy fragment o kolorowym tle (wadą jest to, że nie podlega on podziałowi na strony).

Przykład:

`\noindent\colorbox[named]{Yellow}`

{\begin{minipage}[t]{0.7\columnwidth}

\color{green}

\begin{itemize}

\item tło w kolorze żółtym

\item tło w kolorze żółtym

\item tło w kolorze żółtym

\end{itemize}

\end{minipage}%

}

• tło w kolorze żółtym

• tło w kolorze żółtym

• tło w kolorze żółtym

`\fcolorbox{kolor ramki}{kolor tła}{tekst}`

`\fcolorbox[model]{kolor ramki}`

{kolor tła}{tekst} polecenie powoduje zmianę koloru tła „pudełka” (*tekst* jest

zamykany w konstrukcji `\hbox` zawierającego `tekst`;

Przykład:

```
\fboxrule=0.5mm
\fcolorbox{green}{red}
{tłó w kolorze czerwonym,
w zielonej ramce}
```

tłó w kolorze czerwonym, w zielonej ramce

```
\fboxrule=0.5mm
\fcolorbox[named]{NavyBlue}{Yellow}
{tłó w kolorze żółtym,
w granatowej ramce}
```

tłó w kolorze żółtym, w granatowej ramce

Grubość ramki możemy regulować wartością parametru `\fboxrule`.

Przykład:

```
\fboxrule=2mm
\fcolorbox[named]{NavyBlue}{Yellow}
{tłó w kolorze żółtym,
w granatowej ramce}
```

tłó w kolorze żółtym, w granatowej ramce

natomiast odległość ramki od tekstu, parametrem `\fboxsep`.

Przykład:

```
\fboxsep=3mm
\fboxrule=0.5mm
\fcolorbox[named]{NavyBlue}{Yellow}
{tłó w kolorze żółtym,
w granatowej ramce}
```

tłó w kolorze żółtym, w granatowej ramce

`\pagecolor{ kolor }`

`\pagecolor[model]{ kolor }` polecenie zmienia kolor tła strony dokumentu; **tej na której jest umieszczone** (znaczenie parametrów analogiczne jak poprzednio). Ze względu na inną realizację procesu kolorowania w \TeX u i $\text{pdf}\TeX$ u efekt przetwarzania może być **różny**. W przypadku \TeX a efekt będzie zawsze poprawny (kolor uzyskiwany jest w momencie przetwarzania postscriptem, kiedy znane są już miejsca podziału strony). $\text{pdf}\TeX$ może zrealizować polecenie kolorowania tła przed ostatecznym złamaniem strony, jeżeli znajdzie się ono na końcu akapitu, który

będzie podlegał przełamaniu (w obszarze *recent contributions*).

2.3. Definiowanie kolorów

`\definecolor{ nazwa }{ model }{ definicja }` polecenie umożliwia tworzenie własnego koloru. Parametr *model* może przyjmować następujące wartości: *rgb* – definicje stanowią trzy liczby z przedziału 0-1 (oddzielone przecinkiem), określające składowe R (czerwony), G (zielony) i B (niebieski), np: `\definecolor{blue}{rgb}{0,0,1}`; *cmk* – definicję stanowią cztery liczby z przedziału 0-1 (oddzielone przecinkiem), określające składowe C (cyjan), M (magenta), Y (żółty), K (czarny), np: `\definecolor{yellow}{cmk}{0,0,1,0}`; *gray* – definicję stanowi liczba z przedziału 0-1 określająca nasycenie czerni (1 – kolor biały, 0 – kolor czarny), np: `\definecolor{lekkoszary}{gray}{0.8}`; *named* – definicją może być nazwa jednego z kolorów zdefiniowanych w modelu *named*, np: `\definecolor{yellow}{named}{Yellow}`. Standardowo są zdefiniowane następujące kolory: *black* (czarny), *white* (biały), *red* (czerwony), *green* (zielony), *blue* (niebieski), *cyan*, *magenta*, *yellow* (żółty).

`\DefineNamedColor{ nazwa }`

`{ nowa nazwa }{ model }{ definicja }`

makroinstrukcja do tworzenia kolorów w określonym modelu (w pakiecie jest zdefiniowany jedynie model o nazwie *named*), należy ją traktować jako polecenie „do użytku wewnętrznego”. W wyniku jej działania zostanie utworzone makro – legalna nazwa koloru, ale efekt zmiany koloru będzie możliwy dopiero po odpowiednich modyfikacjach plików dołączanych przez program *dvips*. Standardowo są zdefiniowane kolory przedstawione w tabeli 1:

3. Pakiet colortbl

Pakiet, w połączeniu z wyżej opisanym, umożliwia sterowanie kolorami kolumn, wierszy i pojedynczych cel tabel, oraz zmianę koloru linii oddzielających i koloru wypełnienia między liniami podwójnymi. Aby w pełni wykorzystać jego możliwości należy dołączyć pakiety *array*, *hhline*,

Tabela 1: Tabela kolorów zdefiniowanych w *named color*

GreenYellow	Melon	Mahogany	WildStrawberry	Mulberry
Yellow	YellowOrange	Maroon	Salmon	RedViolet
Goldenrod	Orange	BrickRed	CarnationPink	Fuchsia
Dandelion	BurntOrange	Red	Magenta	Lavender
Apricot	Bittersweet	OrangeRed	VioletRed	Thistle
Peach	RedOrange	RubineRed	Rhodamine	Orchid
DarkOrchid	CornflowerBlue	SkyBlue	Green	Sepia
Purple	MidnightBlue	Turquoise	ForestGreen	Brown
Plum	NavyBlue	TealBlue	PineGreen	Tan
Violet	RoyalBlue	Aquamarine	LimeGreen	Gray
RoyalPurple	Blue	BlueGreen	YellowGreen	Black
BlueViolet	Cerulean	Emerald	SpringGreen	White
Periwinkle	Cyan	JungleGreen	OliveGreen	
CadetBlue	ProcessBlue	SeaGreen	RawSienna	

dcolumn i ewentualnie longtable. W pakiecie są zdefiniowane następujące makroinstrukcje:

```
\columncolor{kolor}[lewa nadlewka]
```

```
[prawa nadlewka]
```

```
\columncolor[model]{kolor}
```

```
[lewa nadlewka][prawa nadlewka]
```

polecenie definiuje kolor tła kolumny tabeli, parametr *model* i *kolor* mają identyczne znaczenie jak przy poprzednio opisywanych poleceniach pakietu *color*. Parametr *lewa nadlewka* (parametr opcjonalny) określa rozmiar rozszerzenia kolorowanego obszaru o podaną wielkość, w celu pokolorowania przestrzeni między separatorem kolumny, a nią samą. Nadanie parametrowi wartości `\tabcolsep` (dla środowiska tabular) ewentualnie `\arraycolsep` (dla środowiska array) spowoduje wypełnienie kolorem całej celi tabeli (jest to wartość domyślna); jeżeli parametr *prawa nadlewka* nie jest określony, obie będą jednakowe (określone wartością parametru *lewa nadlewka*). Użycie polecenia ilustruje przykład, gdzie kolor tła pierwszej kolumny został zdefiniowany jako żółty. Polecenie może być użyte **wyłącznie** przy specyfikacji kolumn (preambuła tabeli lub

polecenie multicolumn) z wykorzystaniem konstrukcji zdefiniowanych w pakiecie *array*.

```
\rowcolor{kolor}
```

`\rowcolor[model]{kolor}` polecenie zmienia kolor (znaczenie parametrów jak w poprzednim poleceniu) tła wiersza tabeli (ma większy priorytet niż zmiana tła kolumny); **musi** być ono umieszczone na początku wiersza.

```
\doublerulesepcolor{kolor}
```

`\doublerulesepcolor[model]{kolor}` polecenie określa kolor wypełnienia między liniami (linia podwójna), zarówno pionowymi, jak i poziomymi.

```
\arrayrulecolor{kolor}
```

`\arrayrulecolor[model]{kolor}` polecenie określa kolor linii.

```
\setlength{\doublerulesep}{wartość}
```

polecenie określa odległość między liniami linii podwójnej.

```
\setlength{\arrayrulewidth}{wartość}
```

polecenie określa grubość linii.

Przykład:

```
\arrayrulecolor{red}
\doublerulesepcolor{blue}
\setlength{\doublerulesep}{0.8mm}
\setlength{\arrayrulewidth}{0.6mm}
\newcolumnntype{V}{>{\columncolor{Violet}}%
```

```

[\tabcolsep]}c}
%definicja kolumny o fioletowym tle
\newcolumntype{G}{>{\color{green}%
\columncolor{Yellow}\tabcolsep}}1}
%definicja kolumny o żółtym tle
%i zielonej czcionce
\begin{tabular}{|G|l|l|}
\hline{t:===:t|}
jeden & dwa & trzy\\
\hline{|:===:|}
cztery & pięć & sześć\\
\hline{|-|-|-|}
\rowcolor{BlueGreen}\color{black}siedem &
\multicolumn{1}{>{\columncolor{Yellow}}c|}%
%wymuszenie koloru pojedynczej celi
{osiem} & dziewięć\\
\hline{|-|-|-|}
dziesięć & \multicolumn{2}{V|}%
%wymuszenie koloru dwóch połączonych celi
{\color{white}jedenaście}\\
\hline{|b:===:b|}
\end{tabular}

```

jeden	dwa	trzy
cztery	pięć	sześć
siedem	osiem	dziewięć
dziesięć	jedenaście	

4. Modyfikacja koloru różnych, typowych elementów dokumentu

4.1. Tytuły

4.1.1. Standardowe klasy dokumentu

Makroinstrukcje definiujące tytuły są tworzone za pomocą sparametryzowanej makroinstrukcji `\startsection`. Ostatnim jej parametrem jest określenie sposobu formatowania samego tytułu (wielkość, rodzaj czcionki itp.) i jest to również właściwe miejsce aby określić kolor liter.

Przykład:

```

\newcounter{msection}
\renewcommand\themsection{\@arabic\c@msection}
\newcommand\msectionmark[1]{
\newcommand\l@msection[2]{
\newcommand\msection{\@startsection%
{msection}{1}{\z@}%
{-0.5ex \@plus -0ex \@minus -.0ex}%
{0.3ex \@plus .0ex}%
{\normalfont\large\bfseries\color{red}}}
\makeatother
\msection{To jest mój czerwony tytuł}

```

brudnopis, 8 września 2004 20:15

1. To jest mój czerwony tytuł

Za sposób prezentacji numeru tytułu odpowiada makroinstrukcja `\@secntformat`, w celu zmiany jego koloru należy ją zmodyfikować.

Przykład:

```

\makeatletter
\renewcommand\@secntformat[1]%
{\color{green}\csname the#1\endcsname.\quad}
%kolor numeru
\makeatother
\msection{To jest mój czerwony
tytuł z zielonym numerkiem}

```

2. To jest mój czerwony tytuł z zielonym numerkiem

W klasach report i book makroinstrukcje `\chapter` są definiowane inaczej (podobnie jak part) i z tego powodu sposób postępowania będzie inny; należy zmodyfikować poniższy kod:

Przykład:

```

\makeatletter
\def\@makechapterhead#1{%
\vspace*{50\p@}%
{\parindent \z@ \raggedright
\normalfont\color{blue}% <---
%kolor numeru
\ifnum \c@secnumdepth >\m@ne
\huge\bfseries \@chapapp\space \thechapter
\par\nobreak
\vskip 20\p@
\fi
\interlinepenalty\@M
\Huge \bfseries\color{red} #1\par\nobreak% <--
%kolor tytułu
\vskip 40\p@
}}
\def\@makeschapterhead#1{%
\vspace*{50\p@}%
{\parindent \z@ \raggedright
\normalfont
\interlinepenalty\@M
\color{blue}% <---
\Huge \bfseries #1\par\nobreak
%kolor tytułu (tytuł nienumerowany)
\vskip 40\p@
}} \makeatother

```

4.1.2. Klasy Marcina Wolińskiego

W tym przypadku, tytuły są tworzone za pomocą makroinstrukcji `\DeclareSectioningCommand`; należy je zmodyfikować wprowadzając deklarację zmiany koloru.

brudnopis, 8 września 2004 20:15

Przykład:

```
\makeatletter
\DeclareSectioningCommand{section}{1}
  {24\p@\@plus5\p@\@minus2\p@}
  {\FormatHangHeading{\large\color{red}}}
  {10\p@\@plus3\p@}
\makeatother
```

Jeżeli wszystkie tytuły mają być prezentowane w tym samym kolorze wystarczy zmienić definicję makroinstrukcji `\sectsetup`.

Przykład:

```
\renewcommand
\sectsetup{\normalfont\bfseries\color{green}}

Za prezentowanie numeru tytułu odpowiada
makro \mw@secntformat i można go zmodyfikować
następująco:
\makeatletter
\renewcommand\mw@secntformat[1]%
  {{\color{green}#1.\quad}}
%oprócz deklaracji zmiany koloru
%dodano nawiasy {}, w celu ograniczenia
%zakresu działania
\makeatother
```

W przypadku polecenia `\chapter` należy zmodyfikować konstrukcję definiującą:

```
\makeatletter
\DeclareSectioningCommand%
  [breakbefore,wholewidth]{chapter}{0}
  {56\p@}
  {\FormatBlockHeading{\LARGE\color{blue}}}
  {24\p@}
\makeatother
```

4.2. Pagina

Sposób prezentowania paginy jest zdefiniowany w standardowych klasach dokumentów. W celu uzyskania innych efektów należy zmodyfikować istniejące makroinstrukcje lub utworzyć własne ewentualnie skorzystać z możliwości jakie daje pakiet `fancyhdr`. Ponieważ pakiet `fancyhdr` nie jest na tyle elastyczny, aby określać kolory tła i czcionki na poziomie „zewnętrznym” (bez modyfikacji kodu pakietu) zdecydowałem się na rozwiązanie problemu w oparciu o standardowe klasy.

4.2.1. Standardowe klasy dokumentów

Poniżej prezentuję skomentowany fragment kodu definiujący paginę górną dla klasy `book` lub `report`

Przykład:

```
\usepackage[normalbf,normalcm]{ulem}
\def\ULthickness{2pt}
%określenie grubości podkreślenia paginy
```

brudnopis, 8 września 2004 20:15

```
\setlength{\ULdepth}{5pt}
%określenie odległości podkreślenia
%od tekstu, powinno być na tyle duże
%aby znalazło się poniżej pudełka
% o zmienionym kolorze tła
\makeatletter
\newdimen\headboxlen
\headboxlen=\textwidth
\advance\headboxlen -2\fbboxsep
\advance\headboxlen -2\fbboxrule
%zdefiniowanie i określenie
%długości pudełka nagłówka
%długość pudełka o zmienionym
%tle jest większa!
\def\ps@mojheadings{%
%definicja własnego stylu nagłówków
\def\@evenhead{\color{blue}%
%kolor linii podkreślającej
  \uline{\colorbox{green}%
%podkreślenie i zdefiniowanie
%koloru tła
{\hbox to \headboxlen{\footnotesize\color{red}%
%zmiana wielkości czcionki i jej koloru
  \thepage\hfil\slshape\leftmark}}}}
\def\@oddhead{\color{blue}
  \uline{\colorbox{green}%
{\hbox to \headboxlen{\footnotesize\color{red}%
%zmiana wielkości czcionki i jej koloru
  {\slshape\rightmark}\hfil\thepage}}}}
%poniżej definicje makroinstrukcji
%wpisującej treść tytułów do
%odpowiednich rejestrów
\def\chaptermark##1{%
\markboth {%
\ifnum \c@secnumdepth >\m@ne
\thechapter. \ %
\fi
##1}{}}%
\def\sectionmark##1{%
\markright {%
\ifnum \c@secnumdepth >\z@
\thesection. \ %
\fi
##1}}
\makeatother
\pagestyle{mojheadings}
%deklaracja używanego stylu
```

brudnopis, 8 września 2004 20:15

Poniżej ilustracja działania użytego kodu

1.1.1. Operowanie matematycznymi wzorami

jei opisanie odpowiednie *modele matematyczne*. Tak postępuje się w fizyce, gdzie np. ruch masy zawieszonyj na nici jest opisywany i analizowany za pomocą modeli matematycznych w postaci *wahadła matematycznego* lub *wahadła fizycznego*. Tak też postępowaliśmy w teorii obwodów, gdzie fizyczne elementy: opornik, cewkę i kondensator opisywaliśmy modelami matematycznymi w postaci elementów obwodowych: *oporu*, *indukcyjności* i *pojemności*. Tak również postępuje się w teorii sygnałów, gdzie sygnały występujące w rzeczywistości opisuje się za pomocą różnego rodzaju modeli matematycznych.

Posługiwanie się modelami matematycznymi sygnałów ma szereg istotnych zalet. Opis sygnału za pomocą modelu matematycznego umożliwia przede wszystkim jego formalną, teoretyczną analizę. Ważną zaletą tego opisu jest także to, iż – w zależności od potrzeb i postawionego celu analizy – temu samemu fizycznemu sygnałowi możemy przyporządkować różne modele o zróżnicowanymi stopniem złożoności. Jeżeli interesują nas jedynie zasadnicze, dominujące cechy sygnału, stosujemy modele prostsze. W przypadku, gdy chcemy w opi-

1.1.2. Pojęcie sygnału.

W znaczeniu potocznym pojęcie *sygnału* jest rozumiane jako proces zmian w czasie pewnej wielkości fizycznej lub stanu obiektu fizycznego. Z tego względu za modele matematyczne sygnałów przyjmujemy funkcje, których argumentem jest czas t . Opisują one ewolucję sygnałów w czasie. W najprostszym przypadku są to funkcje tylko jednej zmiennej t . W przypadkach bardziej złożonych, np. w teorii linii długich lub zagadnieniach przetwarzania obrazów, mogą to być funkcje wielu zmiennych: czasu i współrzędnych przestrzennych. Niekie-

1.1.2 Pojęcie sygnału.

W znaczeniu potocznym pojęcie *sygnału* jest rozumiane jako proces zmian w czasie pewnej wielkości fizycznej lub stanu obiektu fizycznego. Z tego względu za modele matematyczne sygnałów przyjmujemy funkcje, których argumentem jest czas t . Opisują one ewolucję sygnałów w czasie. W najprostszym przypadku są to funkcje tylko jednej zmiennej t . W przypadkach bardziej złożonych, np. w teorii linii długich lub zagadnieniach przetwarzania obrazów, mogą to być funkcje wielu zmiennych: czasu i współrzędnych przestrzennych. Niekie-

Dla klasy `article` należy postąpić analogicznie, zmieniając `chaptermark` i `sectionmark` na:

```
\def\sectionmark##1{%
\markboth {%
\ifnum \c@secnumdepth >\m@ne
\thesection. \ %
\fi
##1}{}}%
\def\subsectionmark##1{%
\markright {%
\ifnum \c@secnumdepth >\z@
\thesubsection. \ %
\fi
##1}}}
```

4.2.2. Klasy Marcina Wolińskiego

W przypadku klasy `mwbk` należy zmodyfikować jedną z makroinstrukcji definiujących paginy, ja wybrałem `uheadings` i na jej podstawie zdefiniowałem:

Przykład:

```
\newdimen\headboxlen
\headboxlen=\textwidth
\advance\headboxlen -2\fbboxsep
\advance\headboxlen -2\fbboxrule
%określenie długości pudełka ze zmienionym tłem
\makeatletter
\renewcommand\ps@mojuheadings{%
%definicja nowego stylu
\def\@oddhead{\vtop{%
\colorbox{yellow}% <---
{\hb@xt@\headboxlen{\color{blue}% <---
%kolor tła żółty, czcionka niebieska
\reset@font{\heading@font\rightmark}}%
\hfil\thepage\strut}}%
\color{red}% <---
\vskip0pt\hrule height 2pt\vskip-2.4pt}}%
%linia oddzielająca czerwona
\let\@oddfoot\@empty
```

```
\def\@evenhead{\vtop{%
\colorbox{yellow}%
{\hb@xt@\headboxlen{\color{blue}% <---
%kolor tła żółty, czcionka niebieska
\reset@font\strut\thepage%
\hfil\heading@font\leftmark}}%
\color{red}\vskip0pt% |—
\hrule height 2pt\vskip-2.4pt}}%
%linia oddzielająca czerwona
\let\@evenfoot\@empty
}
\makeatother
\pagestyle{mojuheadings}
```

4.3. Kolorowanie wzorów matematycznych

Jeżeli nie używamy pakietów $\mathcal{A}\mathcal{M}\mathcal{S}$ -owych zmiana koloru czcionki we wzorach nie nastęrcza kłopotów; wystarczy wywołać dwie makroinstrukcje definiujące akcję przy przejściu w tryb matematyczny: `\everymath` dla trybu tekstowego i `\everydisplay` dla trybu eksponowanego

Przykład:

```
\everymath{\color{red}}
\everydisplay{\color{red}}
to jest  $\sqrt{5}$  wzór nieeksponowany, a to  $\sqrt{5}$  (1)
```

eksponowany.

Kolor tła możemy podłożyć tworząc nowe środowisko `myeq`.

Przykład:

```
\newbox\eqbox
\newdimen\myboxlen
\newenvironment{myeq}
{%
\everymath{\color{Sepia}}
\everydisplay{\color{Sepia}}
\myboxlen=\columnwidth
\fbboxrule=1mm
\advance\myboxlen -2\fbboxsep
\advance\myboxlen -2\fbboxrule
\setbox\eqbox=\vbox\bgroup\hsize\myboxlen%
\abovedisplayskip=0pt%
\abovedisplayshortskip=0pt%
\belowdisplayskip=0pt%
\belowdisplayshortskip=0pt%
\begin{equation}%
}%
{\color{red}}
\end{equation}\egroup%
$$\fcolorbox{blue}{red}{\copy\eqbox}$$%
}
\begin{document}
\begin{myeq}
\sqrt{x}\int_1^{\infty} dx
\end{myeq}%
po
```

i efekt działania³

$$\int_1^{\infty} \sqrt{x} dx \quad (2)$$

po

W przypadku użycia pakietów \mathcal{AMS} -owych i używania **wyłącznie** konstrukcji pochodzących z tych pakietów wystarczy wywołanie makroinstrukcji `everymath` (makro `everydisplay` jest używane do innych celów).

Inaczej również jest wyprowadzany numer wzoru – odpowiada za to makroinstrukcja `\tagform`, którą należy zmodyfikować.

Przykład:

```
\makeatletter
\def\tagform@#1{\maketag@@{\color{blue}%
(\ignorespaces#1\unskip\@italiccorr)}}
\makeatother
\everymath{\color{red}}
\def\dx{\mathop{\texttrm{d}}x}}

\begin{gather}
\int_0^{\infty} x dx
\end{gather}
\end{prz}
```

$$\int_0^{\infty} x dx \quad (3)$$

Dalsze możliwości kolorowania wzorów (w tym tła) daje pakiet `empheq`⁴ [10]; działa on **wyłącznie** z konstrukcjami \mathcal{AMS} - \TeX A.

Przykład:

```
\makeatletter
\everymath{\color{LimeGreen}}
%zielony kolor czcionki
\def\tagform@#1{\maketag@@{\color{LimeGreen}%
(\ignorespaces#1\unskip\@italiccorr)}}
%zielony kolor czcionki numeru wzoru
\makeatother
\def\dx{\mathop{\texttrm{d}}x}}
\begin{empheq}[box=\fcolorbox{blue}{Salmon}]
{gather}
%łososiowe tło z niebieską ramką
\int_0^{\infty} x dx
\end{empheq}
```

³ Z niewiadomej przyczyny polecenie zamykające środowisko należy zakończyć znakiem `%`, w przeciwnym razie na początku następnego wiersza pojawi się dodatkowa spacja.

⁴ Uwaga! opis składni dotyczy wersji 2.11 tego pakietu – w starszych wersjach składnia była inna. Konstrukcje pakietu tej wersji działają poprawnie **wyłącznie** przy przetwarzaniu \TeX -em

$$\int_0^{\infty} x dx$$

(4)

4.4. Standardowe środowiska

Modyfikacja definicji jest prosta i polega na dołączeniu w odpowiednim miejscu wywołania makra `\color`

4.4.1. itemize

Przykład:

```
\makeatletter
\def\itemize{%
\ifnum \@itemdepth >\thr@@\toodeep\else
\advance\@itemdepth\@ne
\edef\@itemitem{%
\labelitem\romannumeral\the\@itemdepth}%
\expandafter
\list
\csname\@itemitem\endcsname
{\def\makelabel##1{\color{red}%
\hss\llap{##1}}}%
%kolor czcionki znaku wyróżniającego
\color{blue}}%
%kolor czcionki tekstu
\fi}
\makeatother
```

- aqq
- bqq
- cqq

4.4.2. enumerate

Podobnie jak poprzednio

Przykład:

```
\makeatletter
\def\enumerate{%
\ifnum \@enumdepth >\thr@@\toodeep\else
\advance\@enumdepth\@ne
\edef\@enumctr{%
\enum\romannumeral\the\@enumdepth}%
\expandafter
\list
\csname label\@enumctr\endcsname
{\usecounter\@enumctr\def\makelabel##1{%
\color{red}\hss\llap{##1}}
%kolor czcionki numeru pozycji
\color{blue}}%
%kolor czcionki tekstu
\fi}
\makeatother
```

1. aqq
2. bqq
3. cqq

4.4.3. description

Przykład:

```
\makeatletter
\renewenvironment{description}
  {\list{}{\labelwidth\z@
    \itemindent-\leftmargin
    \let\makelabel\descriptionlabel}\color{blue}}
  {\endlist}
\renewcommand*{descriptionlabel}[1]%
  {\hspace\labelsep
  \normalfont\bfseries\color{red} #1}
%kolor czcionki tekstu
%kolor czcionki etykiety
\makeatother
```

etykieta 1 aqq

etykieta 2 bqq

etykieta 3 cqq

4.5. Przypisy

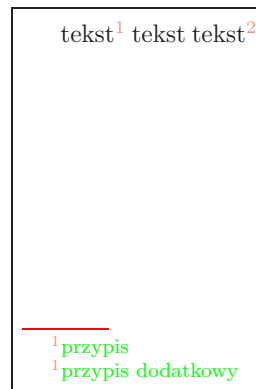
Poniżej przykład ilustrujący możliwość zmiany kolorów tekstu przypisu, linii oddzielającej i jego numeru.

Przykład:

```
\makeatletter
\renewcommand{footnoterule}{%
  \kern-3\p@
  \color{red}\hrule\@width.4\columnwidth
%kolor linii oddzielającej przypisy
  \kern2.6\p@}}
\renewcommand@makefnmark[1]{%
  \parindent 1em%
  \noindent\color{green}
%kolor tekstu
  \hb@xt@1.8em{\hss\@makefnmark}#1}
\def\@makefnmark{\hbox{\@textsuperscript%
  {\normalfont\color{Salmon}\@thefnmark}}}
%kolor numeru przypisu
\makeatother
\pagestyle{empty}
\begin{document}
tekst\footnote{przypis}
tekst\footnotetext{przypis dodatkowy}
tekst\footnotemark
\end{document}
```

efekt działania kodu

brudnopis, 8 września 2004 20:15



4.6. Kolorowe teorie

Oprócz standardowych środowisk, również godna uwagi jest konstrukcja udostępniana przez pakiet amsthm. Makra `\newtheoremstyle` i `\newtheorem` pozwalają na tworzenie nowych środowisk. Parametryczność definicji pozwala również na łatwe wprowadzenie koloru

Przykład:

```
\newtheoremstyle{ctw}%
{0.5\baselineskip}% przed
{0.5\baselineskip}% po
{\normalfont\color{green}}%
%kolor czcionki tekstu
{0pt}%
{\bfseries\color{blue}}%
%kolor czcionki nagłówka
{}%
{\newline}
{{\thmname{#1}}~}%
{\thmnumber{#2}.~}%
{\normalfont\color{cyan}\thmnote{(#3)}}}
%dodatkowa zmiana koloru
%czcionki opcjonalnej części nagłówka
%oraz dołożenie kropki po numerze
\theoremstyle{ctw}
\newtheorem{twierdzenie}{Twierdzenie}
```

i efekt działania:

Twierdzenie 1. (moje twierdzenie)
tekst twierdzenia

$$\sqrt{1} \int_1^{\infty} x dx \quad (5)$$

koniec.

brudnopis, 8 września 2004 20:15

4.7. Spisy

4.7.1. Standardowe klasy

Niestety, podobnie jak w innych standardowych konstrukcjach nie obejdzie się bez modyfikowania kodu L^AT_EX-a. Pozycje w spisie treści (za wyjątkiem chapter w klasie book i report oraz section w klasie article) są definiowane za pomocą makroinstrukcji `\@dottedtocline` i jej kod będziemy modyfikować. W celu zmiany sposobu prezentacji tytułu rozdziału w spisie treści, należy zmodyfikować również makro `\l@chapter` (i analogicznie `\l@section` w klasie article)

Przykład:

```
\makeatletter
\renewcommand*\l@chapter[2]{%
\ifnum \c@tocdepth >\m@ne
\addpenalty{-\@highpenalty}%
\vskip 1.0em \@plus\p@
\setlength\@tempdima{1.5em}%
\begingroup
\parindent \z@ \rightskip \@pnumwidth
\parfillskip -\@pnumwidth
\leavevmode \bfseries
\advance\leftskip\@tempdima
\hskip -\leftskip
{\color{blue}#1}\nobreak\hfil% <---
%kolor tytułu
\nobreak%
\hb@xt@\@pnumwidth{%
\color{red}\hss #2}\par% <---
%kolor numeru strony
\penalty\@highpenalty
\endgroup
\fi}
\def\@dottedtocline#1#2#3#4#5{%
\ifnum #1>\c@tocdepth \else
\vskip \z@ \@plus.2\p@
{\leftskip #2\relax \rightskip \@tocrmarg%
\parfillskip -\rightskip
\parindent #2\relax\@afterindenttrue
\interlinepenalty\@M
\leavevmode
\@tempdima #3\relax
\advance\leftskip \@tempdima%
\null\nobreak\hskip -\leftskip
{\color{blue}#4}\nobreak% <----
%kolor tytułu
{\color{green}\leaders\hbox{\$}\m@th% <----
%kolor kropek
\mkern \@dotsep mu\hbox{.}\mkern \@dotsep
mu$\}\hfill}
\nobreak
\hb@xt@\@pnumwidth{\hfil\normalfont%
\color{red} #5}% <----
%kolor numeru strony
```

brudnopis, 8 września 2004 20:15

```
\par}%
\fi}
\makeatother
```

i wynik przetwarzania

Contents		
1	aqq	3
1.1	sec1	3
1.2	sec2	3
2	bqq	5
2.1	sec1	5
2.2	sec2	5

1

4.7.2. Klasy Marcina Wolińskiego

W tym wypadku wystarczy zmodyfikować jedynie makroinstrukcję `\mw@tocline`

Przykład:

```
\makeatletter
\def\mw@tocline#1#2#3#4#5{%
\ifnum #1>\c@tocdepth \else
\mw@tocskip{#1}{\z@ \@plus.2\p@}%
{\leftskip #3\relax
\@tempdima #3\relax
\advance\@tempdima -#2\relax
\rightskip \@tocrmarg \parfillskip -\rightskip
\interlinepenalty\@M
\noindent\kern-\@tempdima{%
\color{green}\ignorespaces#4}\nobreak% <---
%kolor tytułów
{\color{blue}\leaders\hbox{\$}\m@th% <---
%kolor kropek
\mkern \@dotsep mu\hbox{.}\mkern \@dotsep
mu$\}\hfill}
\nobreak
\hb@xt@\@pnumwidth{\hfil\normalfont%
\color{red} #5}% <---
%kolor numerów stron
\par}%
\fi}
\makeatother
```

i wynik przetwarzania

Spis treści		
Rozdział 1.	aqq	3
1.1.	sec1	3
1.2.	sec2	3
Rozdział 2.	bqq	5
2.1.	sec1	5
2.2.	sec2	5

brudnopis, 8 września 2004 20:15

4.8. Rysunki i tabele (floats)

Jak zwykle wprowadzenie własnych pomysłów wymaga modyfikacji kodu. W poniższym przykładzie chodzi o zmianę koloru podpisu pod rysunkiem (i tabelą).

Za skład podpisu odpowiada makroinstrukcja `\@makecaption`.

Przykład:

```
\makeatletter
\def\figurename{Rys}
%przy okazji zmiana napisu
\long\def\@makecaption#1#2{%
  \vskip\abovcaptionskip
  \footnotesize
%przy okazji wielkości czcionki
  \sbox\@tempboxa{\color{red}\bfseries #1:}%
    \color{green}#2}% <---
%kolor nazwy i opisu dla krótkich tekstów
  \ifdim \wd\@tempboxa >\hsize
    {\color{red}\bfseries #1:} <---
    \color{green}#2\par <---
%kolor nazwy i opisu dla długich tekstów
  \else
    \global \@minipagefalse
    \hb@xt@\hsize{\hfil\box\@tempboxa\hfil}%
  \fi
  \vskip\belowcaptionskip}
\makeatother
```

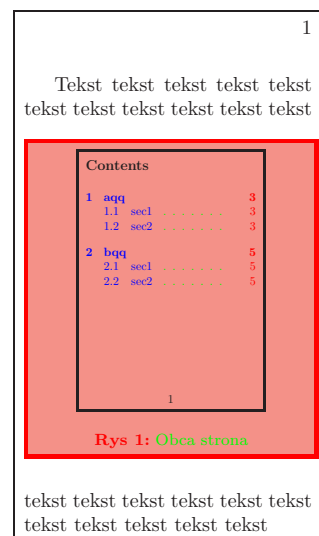
Oczywiście możliwa jest również zmiana tła całego obiektu, można utworzyć konstrukcję analogiczną jak dla wzoru matematycznego z tłem (myeq). Niestety będą trudności z przekazaniem opcjonalnego parametru określającego preferencje umieszczenia rysunku (tabeli)⁵. Poniżej przedstawię prostą konstrukcję działającą bezpośrednio na materiale umieszczonym w środowisku figure.

```
\begin{document}
Tekst tekst tekst tekst tekst tekst tekst
\begin{figure}[h]
\myboxlen=\textwidth
\advance\myboxlen -2\fbboxsep
\advance\myboxlen -2\fbboxrule
\fbboxrule=2pt
\colorbox{red}{\Salmon}{\vbox{\hsize\myboxlen%
\centerline{\fbboxrule=1pt%
\fbbox{\includegraphics%
[width=0.6\columnwidth]{sp.eps}}}}
\caption{Obca strona}}}
\end{figure}%
tekst tekst tekst tekst tekst tekst tekst
tekst tekst tekst tekst tekst tekst tekst tekst

Tekst tekst tekst tekst tekst tekst tekst
\end{document}
```

⁵ Oczywiście z tym problemem też można sobie poradzić, ale nie chciałbym zaczynać pisania L^AT_EXa od początku.

poniżej efekt działania przedstawionego kodu



Literatura

- [1] American Mathematical Society: Dokumentacja pakietu amsthm
- [2] D.Arseneau: Dokumentacja pakietu ulem
- [3] D.Carlisle: Dokumentacja pakietu colortbl
- [4] D.Carlisle: Dokumentacja pakietu color
- [5] D.Carlisle: Dokumentacja pakietu dcolumn
- [6] D.Carlisle: Dokumentacja pakietu hhline
- [7] D.Carlisle: Dokumentacja pakietu longtable
- [8] M.Gosens, F.Mittelbach i A.Samarin: The LaTeX Companion, (ISBN 0-201-54199-8)
- [9] J.Hefferon: Dokumentacja pakietu shadethm
- [10] M.Høgholm: The empheq package, dokumentacja pakietu empheq
- [11] W.Macewicz: Opis pakietu amsthm, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/amsthm.html>
- [12] W.Macewicz: Opis pakietu ulem, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/ulem.html>
- [13] W.Macewicz: Opis pakietu colortbl, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/colortbl.html>
- [14] W.Macewicz: Opis pakietu color, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/color.html>
- [15] W.Macewicz: Opis pakietu dcolumn, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/dcolumn.html>
- [16] W.Macewicz: Opis pakietu array, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/array.html>
- [17] W.Macewicz: Opis pakietu hhline, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/hhline.html>
- [18] W.Macewicz: Opis pakietu longtable, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/macro/longtable.html>
- [19] W.Macewicz: Tytuły w dokumencie, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/tytuły/s.html>
- [20] W.Macewicz: Listy (wykazy), <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/listy/list.html>
- [21] W.Macewicz: Spisy w LaTeXu, <http://www.ia.pw.edu.pl/~wujek/tex/latex2e/spisy/spis.html>
- [22] F.Mittelbach: Dokumentacja pakietu array
- [23] Standardowe klasy dokumentów L^AT_EX2e
- [24] M.Woliński: Moje własne klasy dokumentów dla L^AT_EXa