

Piotr Pianowski
Piotr Strzelczyk

No to

CMYK

Wydanie I
BACHOTEK 2000

1. Wstęp

Podczas spotkania w Bachotku w 1995 r. zaprezentowaliśmy pakiet CMYK-HAX makr T_EX-owych służący do pracy z kolorami, a w szczególności, do tworzenia wyciągów barwnych w T_EX-u. Teraz, po pięciu latach doświadczeń i ulepszeń, chcielibyśmy przekazać do użytkownika nową wersję CMYK-HAX-a. W istotnym stopniu wykorzystujemy w nim cechy poziomu drugiego POSTSCRIPT-u, zwłaszcza w nowopowstałym fragmencie przetwarzającym mapy bitowe.

2. Kolorowa grafika i tekst

Wspinałym i niezawodnym połączeniem T_EX-a z POSTSCRIPT-em jest sterownik DVIPS autorstwa Tomasa Rokickiego. W standardowym zestawie umożliwia on łączenie T_EX-a z grafiką oraz tworzenie kolorowego składu poprzez barwienie tekstu.

Kolorową grafikę do tekstu włącza się tak samo jak szarą, czy czarno-białą. Służy do tego celu pakiet makr EPSF.TEX, stanowiący część oprogramowania dystrybuowanego wraz ze sterownikiem DVIPS. Sposób posługiwania się tymi makrami został opisany w Bachotkowej pracy pt. *Korzystanie z grafiki POSTSCRIPT-owej w dokumentach T_EX-owych*. Nieco lepszym rozwiązaniem jest zastosowanie pakietu EPSFX.TEX – opracowanego w firmie BOP rozszerzenia podstawowego pakietu. Polecenie `\eps` zawarte w tym pakiecie umożliwia wstawianie dowolnych EPS-ów, w tym generowanych przez DVIPS-a, plików zawierających w nazwach niektóre znaki specjalne, a także usuwa niemiłe ograniczenie wielkości obrazka do ok. 50 cm.

Pakiet makr COLORDVI.TEX, także będący częścią zestawu dystrybucyjnego sterownika DVIPS, pozwala kolorować tekst, linie T_EX-owe (ang. *rule*) oraz jednobarwne mapy bitowe (`imagemask`). Aby wyróżnić napis kolorem czerwonym wystarczy napisać: `\Red{tekst czerwony}`. Nazwy kilkudziesięciu kolorów (w języku angielskim) stanowią gotowe makra barwiące. Pełną listę predefiniowanych kolorów można znaleźć w prologu POSTSCRIPT-owym COLOR.PRO.

Pakiet COLORDVI.TEX definiuje kolory posługując się modelem CMYK, najpowszechniej używanym w drukarstwie. Nazwa CMYK pochodzi od angielskich nazw barw podstawowych w tym modelu: *cyan* — niebiesko-zielony, *magenta* — karmazynowy, *yellow* — żółty, *black* — czarny.

W modelu CMYK kolory definiuje się poprzez podanie czterech liczb z przedziału od 0 do 1, wyznaczających poziomy wysycenia kolejnych kolorów podstawowych. Tak więc np. kolor czerwony zdefiniowany jest jako czwórka 0 1 1 0 — 100% magenty i 100% żółtego, OliveGreen (oliwkowy — predefiniowany w COLORDVI.TEX) ma składowe 0.64 0 0.95 0.40, czyli 64% cyjanu, 95%

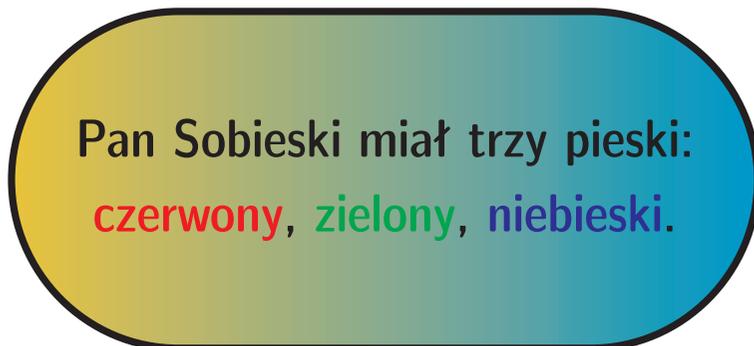
żółtego i 40% czerni. Pakiet `COLORDVI.TEX` pozwala użyć dowolnego koloru o składowych wyrażonych w modelu CMYK. Możemy napisać `\Color{0.64 0 0.95 0.40}{tekst do pokolorowania}` zamiast `\OliveGreen{tekst do pokolorowania}`.

Podane wyżej przykłady definiują kolor lokalnie, dla parametru makra. Można również ustawić bieżący kolor, którego obszar obowiązywania ustalają w zwykły sposób reguły grupowania \TeX -owego. Makro `\textColor{<składowe koloru>}`, a także seria makr związana z predefiniowanymi kolorami `\textRed`, `\textOliveGreen`, itd., definiują bieżący kolor.

Następujący przykład prostego programu \TeX -owego ilustruje powyższe uwagi:

```
\input epsfx
\input colordvi
\nopagenumbers
\font\bf plssdc10 at 20pt \bf
\baselineskip 28pt
\hsize 100mm
% wkładamy do pudełka nr 1 kolorową ramkę:
\setbox1 \hbox{\epsfxsize\hsize \epsffile{fountain.eps}}
% na kolorową ramkę nakładamy kolorowy tekst:
\hbox{\rlap{\copy1}\vbox to \ht1{\vss
  \textBlack
  \centerline{Pan Sobieski miał trzy pieski:}
  \centerline{\Red{czerwony}, \Green{zielony},
    \Blue{niebieski}.}
  \vss}}
\end
```

Program ten generuje następujący rysunek:



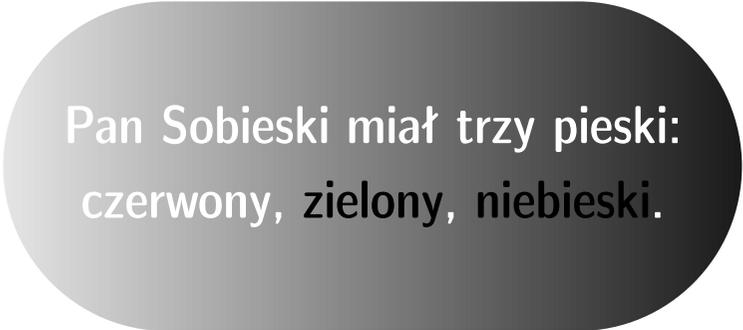
Jaka szkoda, że Państwo nie widzą tych kolorów!

3. Separacje kolorów

Dokument T_EX-owy zawierający barwne elementy można obejrzeć na kolorowym ekranie lub wydrukować na kolorowej drukarce atramentowej. Drukarnia offsetowa na ogół oczekuje od nas jednak rozbarwień, to znaczy wykonanych na fotonaświetlarce diapozytywów odpowiadających barwom podstawowym w modelu CMYK.

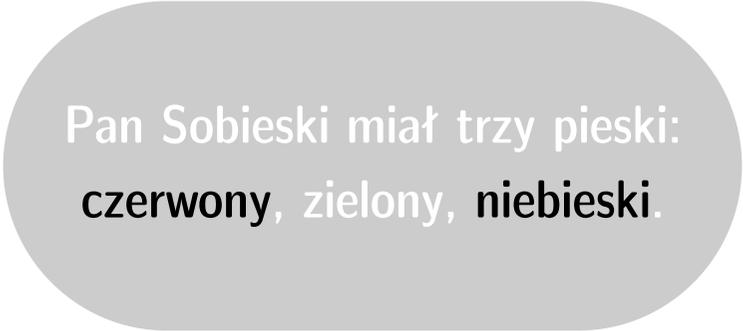
Zestaw makr T_EX-owych o nazwie CMYK-HAX.TEX ułatwia wykonywanie separacji kolorów.

Powyższy (w rzeczywistości barwny) rysunek po separacji barw zamienia się na cztery czarno-białe rysunki.



Pan Sobieski miał trzy pieski:
czerwony, zielony, niebieski.

CYAN



Pan Sobieski miał trzy pieski:
czerwony, zielony, niebieski.

MAGENTA

**Pan Sobieski miał trzy pieski:
czerwony, zielony, niebieski.**

YELLOW

Pan Sobieski miał trzy pieski:

, , .

BLACK

Powyższe ilustracje zostały uzyskane po wykonaniu niewielkich modyfikacji przedstawionego wyżej programu T_EX-owego, wykorzystujących CMYK-HAX. Pierwsza rzecz to wczytanie zbioru makr (instrukcja `\input cmyk-hax` umieszczana bywa zwykle na początku dokumentu), a następnie użycie jednej z następujących instrukcji:

```
\projectCMYK\cyan
\projectCMYK\magenta
\projectCMYK\yellow
\projectCMYK\black
```

właściwej dla danego wyciągu barwnego.

Fachowcy powiedzą przypuszczalnie, że te separacje są „nietechnologiczne”. Na wszystkich wyciągach, z wyjątkiem czarnego pozostały białe „duchy” w miejscu czerni. Istnieje duże niebezpieczeństwo, że na gotowych drukach pojawią się białe szczeliny przylegające do czarnego koloru. Prawie niemożliwe staje

się wydrukowanie tak przygotowanego dokumentu bez tego niekorzystnego efektu.

Ponieważ kolor czarny jest dominujący, w drukarstwie stosuje się tzw. nadruk (ang. *overprint*), czyli drukowanie czernią na pozostałych kolorach.

Makra CMYK-HAX umożliwiają to podejście, poprzez m. in. pomijanie obiektów czarnych na innych wyciągach. W tym celu instrukcję `\projectCMYK` należy poprzedzić podstawieniem `\delblacktrue`, powodującym „znikanie” czarnych obiektów ze wszystkich wyciągów, oprócz czarnego.

Dokładniejszemu opisowi makr tu użytych i ich możliwościom jest poświęcony rozdział 5 niniejszego opracowania.

Oto wynik „zignorowania” czerni na trzech wyciągach.



czerwony zielony niebieski

CYAN



czerwony zielony niebieski

MAGENTA



czerwony zielony niebieski

YELLOW

Wyciąg czwarty (BLACK) nie ulega zmianie.

4. Kolorowe mapy bitowe

Zadanie tworzenia wyciągów barwnych dokumentów zawierających ilustracje w postaci kolorowych map bitowych znajduje swoje rozwiązanie w pakiecie CMYK-HAX. Możliwość ta została wprowadzona do pakietu CMYK-HAX od wersji 0.60. Obecna wersja CMYK-HAX-a (0.70) współpracuje z, praktycznie, dowolnymi mapami bitowymi, tj. `colorimage`, `image` i `imagemask`. Wymaga jednak użycia interpretera `POSTSCRIPTLevel 2`, ze względu na zastosowaną wewnętrzną notację map bitowych. W tej wersji mapy bitowe podlegają operacji `\projectCMYK`. Natomiast operacje takie jak: `\changeColor`, `\delcolor` itp. mają zastosowanie tylko do jednokolorowych map bitowych (`imagemask`). Podmiana kolorów jest możliwa także w mapie bitowej szarej lub kolorowej poprzez zastosowanie operacji `\forbitmap` i `\changebitmap` — patrz niżej.

Pakiet obsługuje mapy bitowe spełniające następujące warunki: `ImageType = 1`, model barw należący do klasy `Device Color Spaces`, `BitsPerComponent = 8` (inne wartości `BitsPerComponent` dopuszczalne są dla `imagemask` (oczywiście 1) oraz w przypadku modelu `DeviceGray` — bez możliwości podmiany koloru). W praktyce wszystkie pliki generowane przez współczesne programy graficzne spełniają te warunki.

Włączanie pełnokolorowych plików do dokumentu rozbarwień, może powodować „puchnięcie” plików `POSTSCRIPT`-owych. Rozwiązaniem jest stosowanie w dokumencie `TEX`-owym `EPS`-ów nagłówkowych, odwołujących się do zewnętrznych plików `TIF` lub `EPS`, które zawierają właściwy rysunek. Takie podejście umożliwiają programy `TIFF2PS` oraz `HEPS` stworzone w firmie `BOP` (pakieci

HEPS tworzący pliki nagłówkowe do EPS-ów dostępny jest jedynie podczas konferencji BachoT_EX, a pakiet TIFF2PS można znaleźć także w internetowych archiwach GUST-u).

Możliwe jest również inne podejście do rozbarwiania bitmap, które praktycznie nie wykorzystuje CMYK-HAX-a. Zakładamy w nim bowiem, że główna robota — rozbarwienie map bitowych (np. zdjęć) i przygotowanie rozbarwień dla poszczególnych kolorów w postaci plików EPS — została wykonana przy użyciu specjalnych programów graficznych (można do tego celu użyć wspomnianego wyżej programu TIFF2PS). Po wykonaniu tej części pracy reszta jest prosta.

Przypuśćmy, że w tekście mamy umieścić jedno tylko zdjęcie; przypuśćmy dalej, że pliki Z.EPS, Z-C.EPS, Z-M.EPS, Z-Y.EPS oraz Z-K.EPS zawierają barwną mapę bitową zdjęcia (Z.EPS) oraz wyciągi barwne (odpowiednio *cyan*, *magenta*, *yellow*, *black* w plikach Z-C.EPS, Z-M.EPS, Z-Y.EPS, Z-K.EPS).

Aby w pliku POSTSCRIPT-owym tworzonym przez DVIPS znalazł się odpowiedni dla danej separacji wyciąg, można użyć następującej konstrukcji T_EX-owej:

```
\sepbitmapfalse
\epsffile{\ifx\projectcolor\cyan Z-C.EPS
\else \ifx\projectcolor\magenta Z-M.EPS
\else \ifx\projectcolor\yellow Z-Y.EPS
\else \ifx\projectcolor\black Z-K.EPS
\else Z.EPS \fi\fi\fi\fi}
```

Powyższa konstrukcja wykorzystuje fakt definiowania przez instrukcję `\projectCMYK` makra `\projectcolor`, które wyznacza bieżący kolor separacji; Po wyłączeniu separowania bitmap (`\sepbitmapfalse`), używa się, po prostu, EPS-a z odpowiednim wyciągiem.

Niektóre programy graficzne (np. PHOTOSHOP) generują samorozbarwiające się EPS-y, które zawierają wewnątrz coś w rodzaju powyższej konstrukcji. W zależności od bieżącego koloru wyciągu plik rozwijają się do odpowiedniej szarej bitmapy. Gdy pracujemy w trybie pełnokolorowym, bitmapa jest kolorowa. Także w tym przypadku należy wyłączyć rozbarwianie map bitowych przez pakiet CMYK-HAX – podstawiając `\sepbitmapfalse`.

5. Opis makr CMYK-HAX

Na wstępie parę uwag ogólnych.

- Tekst jest przez POSTSCRIPT traktowany podobnie jak obwiedniowe obiekty graficzne. Różnica dotyczy tekstów reprezentowanych jako mapy bitowe i polega na niemożności obrysowania brzegów znaku — nie wiadomo, jak należałoby w takim przypadku rozumieć zabieg obrysowywania.

- T_EX-owe linie (`\hrule` i `\vrule`), przeniesione do POSTSCRIPT-u przez sterownik DVIPS, w oryginalnej postaci nie znikają, mimo stosowania `\delblacktrue`. Powodem jest bardzo trikowy sposób ich zdefiniowania. Dołączony do pakietu CMYK-HAX plik RULES.TEX zawiera makro, którego umieszczenie na każdej stronie dokumentu usuwa tę niedogodność (por. następna uwaga).
- Sterownik DVIPS ma tę cechę, że jedynie definicjom umieszczanym w nagłówku gwarantuje globalność. Zasięg pozostałych definicji jest ograniczony do pojedynczej strony. Dokładniej, kod POSTSCRIPT-owy każdej strony rozpoczyna operacja `save`, zapamiętująca bieżący stan definicji, a kończy instrukcja `restore`, odtwarzająca stan poprzedni.

Chcąc używać makr z zestawu CMYK-HAX globalnie, należy programować w T_EX-u na poziomie procedury wyjściowej (ang. *output routine*). Brzmi to groźnie, ale sprowadza się w praktyce do konstrukcji postaci `\headline{\langle instrukcje zestawu CMYK-HAX \rangle\langle właściwa definicja nagłówka strony, być może pusta \rangle}`.

- Podstawowy mechanizm implementacji zasada się na przedefiniowaniu standardowego znaczenia operacji POSTSCRIPT-owych: `fill`, `eofill`, `stroke` (dotyczących obiektów graficznych), `show` (dotyczącej tekstów), `image`, `colorimage`, `imagemask` (dotyczących map bitowych) oraz `setcmykcolor` (dotyczącej tak obiektów graficznych jak i tekstów).

Szczegóły można wprawdzie znaleźć w kodzie źródłowym, tym niemniej, chcąc lepiej kontrolować rezultaty, warto wiedzieć, że przedefiniowanie operacji `fill`, `eofill` i `stroke` polega w istocie na „rozmnożeniu” instrukcji: treścią przeddefiniowanej instrukcji `fill` jest — oprócz instrukcji związanych ze sprawdzeniem aktualnego koloru — opcjonalne wykonanie kolejno oryginalnych operacji `stroke`, `fill` i ponownie `stroke`. Podobnie operacja `eofill` jest zamieniana na `stroke`, `eofill` i `stroke`, zaś operacja `stroke` — na `stroke`, `fill` i `stroke`. Z pojawiających się podwójnych operacji `stroke` wykonywana jest tylko jedna, w zależności od użycia wcześniej instrukcji `\strokeup` bądź `\fillup`.

- Ukłonem w stronę POSTSCRIPT-u jest forma specyfikowania kolorów: w makrach T_EX-owych zestawu CMYK-HAX jest to pojedynczy parametr zawierający cztery liczby z zakresu od 0 do 1 separowane odstępami, za wyjątkiem kolorów do podmiany w modelach barw RGB lub Gray, które reprezentowane są odpowiednio jako trójki liczb lub pojedyncze liczby.

A oto lista najważniejszych makr T_EX-owych zestawu CMYK-HAX.

Instrukcje `\PSbegingroup` i `\PSENDGROUP`

Jest to para makr-nawiasów analogicznych do T_EX-owych instrukcji `\begingroup` i `\endgroup`, tyle że mająca też wpływ na wynikowy kod POSTSCRIPT-owy: zasięg działania makr zestawu CMYK-HAX jest ograniczony do obszaru pomiędzy `\PSbegingroup` a `\PSENDGROUP`. Nawiasy te są użyteczne wówczas gdy na jednej stronie chcemy umieścić kilka wariantów rozbarwień lub zmian kolorów.

Uwaga: Użytkownik sam musi zadbać o to by para makr `\PSbegingroup` i `\PSENDGROUP` znalazła się na tej samej stronie. Jest to konieczne ze względu na strukturę wynikowego pliku POSTSCRIPT-owego generowanego przez DVIPS — por. uwagi wstępne do niniejszego rozdziału dotyczące globalnych i lokalnych definicji. Najbezpieczniej jest używać pary instrukcji `\PSbegingroup` i `\PSENDGROUP` wewnątrz pudełka (`\hbox-a` lub `\vbox-a`).

Użycie makr `\PSbegingroup` i `\PSENDGROUP` związane jest z jeszcze jednym niebezpieczeństwem — ze względu na interesującą przyrostową metodę definiowania odstępów w plikach generowanych przez DVIPS-a, mogą występować dziwne anomalie w pozycjonowaniu tekstu. Jeśli pierwszy odstęp między słowami (spacja) na stronie znajduje się wewnątrz grupy POSTSCRIPT-owej, to wszystkie spacje poza tą grupą znikną. Jest to bardzo niebezpieczna i trudna do uniknięcia właściwość tych makr.

Instrukcja `\projectCMYK`

Jest to podstawowa instrukcja do tworzenia wyciągów barwnych. Dopuszczalne są następujące warianty (por. rozdz. 3): `\projectCMYK \cyan`, `\projectCMYK \magenta`, `\projectCMYK \yellow` oraz `\projectCMYK \black`.

Są to instrukcje przewidziane raczej do zabiegów globalnych, toteż — jak to zostało powiedziane wyżej — powinny być umieszczane w nagłówku strony (`\headline`).

Rozważmy fragment składu, gdzie bieżący kolor ma składowe (c, m, y, k) . Efekt działania instrukcji `\projectCMYK \cyan` jest w tym obszarze równoważny przestawieniu bieżącego koloru na $(0, 0, 0, c)$; analogicznie efekt działania instrukcji `\projectCMYK \magenta`, `\projectCMYK \yellow`, `\projectCMYK \black` jest równoważny odpowiednio przestawieniu bieżącego koloru na $(0, 0, 0, m)$, $(0, 0, 0, y)$, $(0, 0, 0, k)$.

Instrukcja `\projectCMYK` potrafi rozbarwić również bitmapy jednokolorowe (`\imagemask`), szare, kolorowe (zarówno w modelu CMYK, jak i RGB), a także barwione za pomocą makr pakietu `COLORMAP.TEX`. Wykonywanie rozbarwienia mapy bitowej można zablokować, ustalając `\sepbitmapfalse`.

Instrukcja `\projectCMYK` ustawia też liniaturę (gęstość i kąt rastra oraz funkcję plamki). Domyślnie przyjęliśmy gęstość rastra równą 150 linii na cal (odpowiednią dla rozdzielczości fotonaświetlarki 2400 dpi), definiując `\def\ScreenFrequency {150}`; kąt rastra określony jest za pomocą makr `\def\Cangle {15}`, `\def\Mangle {75}`, `\def\Yangle {0}`, `\def\Kangle {45}`, a funkcja plamki odpowiada najpowszechniej stosowanemu rastrowi kołowemu (jak na okładce tego zeszytiku). Użytkownik, wiedzący co robi, może zdefiniować te makra inaczej, zmieniając liniaturę.

Pozostałe makra przeznaczone są do tworzenia specjalnych wyciągów barwnych w sytuacjach, gdy zachodzi konieczność zmiany koloru, dodania lub zmiany obrysu, itp.

Instrukcje `\setCMYKchange` i `\useCMYKchange`

Instrukcja `\setCMYKchange` powinna poprzedzać wszystkie definicje związane z podmianą kolorów. Jej zadaniem jest w zasadzie inicjalizacja odpowiednich makr T_EX-owych. Żaden kod POSTSCRIPT-owy nie jest wpisywany do pliku DVI.

Instrukcja `\useCMYKchange` zamyka listę podmian kolorów i wpisuje do pliku DVI odpowiedni kod POSTSCRIPT-owy. Dopiero od tego momentu zmiany wprowadzone między wystąpieniem instrukcji `\setCMYKchange` i instrukcji `\useCMYKchange` mają wpływ na obiekty umieszczane w składzie.

Instrukcja `\forcolor`

Jest to jednoparametrowe makro ustalające bieżący kolor, wykorzystywane przez makra `\change fill`, `\changestroke`, `\change color`, `\del fill`, `\del stroke`, `\del color` `\add fill`, `\add stroke`, `\add color`, `\fill hook`, `\stroke hook`, `\fill up` oraz `\stroke up`. Parametr makra `\forcolor` winien rozwijać się do poprawnej dla POSTSCRIPT-u czwórki wartości opisujących kolor w modelu CMYK. W wyniku działania operacji `\forcolor` zostają zdefiniowane POSTSCRIPT-owe wielkości `c_`, `m_`, `y_` oraz `k_`, które są równe czterem składowym koloru (w modelu CMYK), i mogą być wykorzystane w trakcie podmiany koloru.

Instrukcje `\change fill`, `\changestroke` i `\change color`

Są to jednoparametrowe makra określające kolor jakim ma zostać podmieniony kolor wypełnienia (`\change fill`) bądź obrysu (`\changestroke`) określony przez bezpośrednio poprzedzającą ich użycie instrukcję `\forcolor`. Powinny one zostać umieszczone po użyciu instrukcji `\setCMYKchange`, a przed użyciem instrukcji `\useCMYKchange`.

Instrukcja `\change color` jest równoważna równoczesnemu użyciu instrukcji `\change fill` i `\changestroke`.

Instrukcje `\delfill`, `\delstroke` i `\delcolor`

Są to bezparametrowe instrukcje powodujące, że bieżący kolor ustalony przez makro `\forcolor` będzie ignorowany w przypadku obiektu wypełnianego (`\delfill`) bądź obrysowywanego (`\delstroke`). Innymi słowy POSTSCRIPT pominie operacje odpowiednio `fill` (lub `eofill`) i `stroke`.

Podobnie jak instrukcje wymienione w punkcie poprzednim, powinny one zostać umieszczone po użyciu instrukcji `\setCMYKchange`, a przed użyciem instrukcji `\useCMYKchange`.

Instrukcja `\delcolor` jest równoważna równoczesnemu użyciu instrukcji `\delfill` i `\delstroke`.

Instrukcje `\addstroke`, `\addfill`, `\addcolor`

Instrukcje te dodają do obiektu wypełnianego obwódkę (`\addstroke`), a do obiektu obrysowywanego — wypełnienie (`\addfill`). Kolor dodanej obwódki bądź wypełnienia podaje się jako parametr.

Podobnie jak instrukcje wymienione w punkcie poprzednim powinny one zostać umieszczone po użyciu instrukcji `\setCMYKchange`, a przed użyciem instrukcji `\useCMYKchange`.

Instrukcja `\addcolor` jest równoważna równoczesnemu użyciu instrukcji `\addfill` i `\addstroke`.

Instrukcje `\fillhook` i `\strokehook`

Są to jednoparametrowe instrukcje definiujące operacje POSTSCRIPT-owe, które należy wykonać przed wypełnieniem obiektu (`\fillhook`) bądź obrysowaniem (`\strokehook`). Ich treścią jest po prostu kod POSTSCRIPT-owy. Typowo wykorzystuje się makro `\strokehook` do ustalenia grubości linii (operacja POSTSCRIPT-owa `setlinewidth`), kształtu zakończeń (`setlinecap`) i kształtu naroży, w przypadku niegładkiej zmiany kierunku krzywej (`setlinejoin` i `setmitterlimit`). Na przykład użycie instrukcji

```
\strokehook{2 trueemm setlinewidth
  1 setlinecap
  0 setlinejoin}
```

spowoduje, że obwódka będzie miała grubość dwóch milimetrów, jej naroża będą zaokrąglone, a końce ostro ścięte w węzłach ścieżki. To wszystko odnosi się oczywiście do bieżącego koloru ustalonego przez `\forcolor`.

Dwie uwagi na temat kodu POSTSCRIPT-owego występującego w przykładowym użyciu makra `\strokehook`.

1. Pamiętać należy, że zgodnie z regułami POSTSCRIPT-u argumenty operacji podawane są przed wystąpieniem samej operacji. Reguła taka nosi nazwę odwrotnej notacji polskiej i umożliwia podawanie parametrów bez użycia nawiasów.

W związku z tym powyższy zapis należy interpretować następująco: liczba 2 jest argumentem instrukcji `truemm` (p. następna uwaga), wynik jej działania (liczba) jest argumentem operacji `setlinewidth`, liczba 1 jest argumentem operacji `setlinecap`, liczba 0 jest argumentem operacji `setlinejoin`.

2. Operacja `truemm` jest makrem *POSTSCRIPT*-owym zdefiniowanym w zestawie *CMYK-HAX*. Oprócz niego — jak można by się spodziewać — zostały również zdefiniowane makra `truebp`, `truept`, `trueesp`, `truecm`, `truein`, `truedd` oraz `truecc`. Wyliczają one odpowiedni parametr — na ogół dla operacji `setlinewidth`, a tym samym również dla operacji `stroke` — sprawdzając aktualnie obowiązujące przeskalowanie (macierz bieżącego przekształcenia).

Instrukcje `\fillup` i `\strokeup`

Instrukcje te służą do ustalenia czy obwódka ma się znaleźć pod wypełnieniem (`\fillup`), czy też wypełnienie pod obwódką (`\strokeup`). Oczywiście decyzja taka ma znaczenie jedynie w przypadku obrysowywania obiektu graficznego innym kolorem niż kolor wypełnienia.

Instrukcja `\forbitmap`

Jest to jednoparametrowe makro ustalające bieżący kolor mapy bitowej, wykorzystywany przez makro `\changebitmap`. Jeśli bitmapa należy do modelu *CMYK*, to parametrem jest czwórka dwucyfrowych liczb szesnastkowych, tzn. liczb z zakresu 0–255, zapisanych z użyciem znaków: 0, 1, ... 9, A, B, C, D, E, F, koniecznie z zerami wiodącymi, wewnątrz mogą występować spacje, małe i duże litery są dopuszczalne. Liczby te reprezentują piksel mapy bitowej, a dokładniej — jego kolor w modelu *CMYK*. W przypadku, gdy mapa bitowa należy do modelu *RGB*, wówczas parametrem jest trójka takich liczb, a dla szarej bitmapy – pojedyncza liczba. Użytkownik odpowiedzialny jest za właściwą postać parametru.

Instrukcja `\changebitmap`

Jest to jednoparametrowe makro określające jakim kolorem ma być w przetwarzanej mapie bitowej podmieniony kolor ustalony przez makro `\forbitmap`. Parametrem jest czwórka (trójka, jedynka) dwucyfrowych liczb szesnastkowych (tak jak w makrze `\forbitmap`).

6. Uwagi końcowe

Ewentualnych użytkowników zestawu makr *CMYK-HAX* należy uprzedzić, że zabawy z kolorami są trudne, i to nie z powodu *T_EX*-a, a z powodu natury problemu. Tym niemniej umiejętność operowania kolorami to podstawa nowoczesnego składu tekstów. Narzędzie *T_EX+POSTSCRIPT* idealnie pasuje do tego problemu. Proszę spróbować — naprawdę warto.

SPIS TREŚCI

	str.
1. Wstęp	1
2. Kolorowa grafika i tekst.....	1
3. Separacje kolorów	3
4. Kolorowe mapy bitowe.....	6
5. Opis makr CMYK-HAX.....	7
6. Uwagi końcowe	12

