

```
3. pdf(e)la(tex) jobname % Here thumbpdf.sty
    % is required
```

◇ Heiko Oberdiek
oberdiek@ruf.uni-freiburg.de

8.4. Other images as thumbnails

Other images, symbols, or pictograms in the formats PNG, TIFF, or JPEG can be used as thumbnails instead of the reduced image of the page. If some symbols or pictograms should be included instead of reduced images, they can be specified with `\thisthumb` in the document \TeX file. In the project `thumbpdf` the `thumbpdf.tex` picks up the thumbnail images. The extra symbols are announced to `thumbpdf.tex` by declaring them in the file `thumbopt.tex` that will be scanned, if it exists. This optional file consists of `\thumb` macros with an optional label and the filename as arguments. Although PNG, TIFF, and JPEG files can be used, only the file extension `.png` can be omitted.

The names, used in references later, can be the filenames themselves or labels, declared in the optional argument of `\thumb`:

<code>thumbopt.tex</code>		<code>jobname.tex</code>
<code>\thumb{one}</code>	→	<code>\thisthumb{one}</code>
<code>\thumb{one.png}</code>	→	<code>\thisthumb{one.png}</code>
<code>\thumb[two]{one}</code>	→	<code>\thisthumb{two}</code>
<code>\thumb[one]{./extras/one.png}</code>	→	<code>\thisthumb{one}</code>

9. Links

- PDF specification (version 1.3):
<http://partners.adobe.com/asn/developer/PDFS/TN/PDFSPEC.PDF>
- pdf \TeX (Hàn Thê Thành):
<http://www.tug.org/applications/pdftex/>
Win32: <ftp://ftp.ese-metz.fr/pub/TeX/win32-beta/>
- hyperref (Sebastian Rahtz):
<ftp://ctan.tug.org/tex-archive/macros/latex/contrib/supported/hyperref/>
If test versions are available, then they can be get from
<http://www.tug.org/applications/hyperref/hyperref.zip>
- hypbmsec (Heiko Oberdiek):
<ftp://ctan.tug.org/tex-archive/macros/latex/contrib/supported/oberdiek/>
- thumbpdf (Heiko Oberdiek):
<ftp://ctan.tug.org/tex-archive/macros/pdftex/thumbpdf/>

PDF

(La) \TeX i PDF: fonty i grafika

Wojciech Myszka

1. Wstęp

Format PDF cieszy się (zasłużenie) dużym zainteresowaniem wśród wszystkich, którzy zajmują się przygotowaniem dokumentów elektronicznych. W porównaniu z dokumentami HTML, dokumenty PDF mają doskonałą postać graficzną umożliwiającą ich wydruk w założonej przez autora formie.

Dokument PDF może być także zabezpieczony hasłem przed możliwością drukowania czy ekstrahowania z niego ilustracji i/lub tekstu.

W porównaniu z formatem PS, dokumenty PDF są zazwyczaj mniejsze (dzięki zastosowaniu kompresji i różnych, zaawansowanych technik przetwarzania ilustracji). Podobnie jak pliki PS, dobrze przygotowane dokumenty PDF są „skalowalne”, to znaczy operacje powiększania i zmniejszania nie powodują utraty ich jakości.

Podobnie jak dokumenty HTML, pliki PDF są w pełni hipertekstowe; dzięki udostępnionym przez firmę Adobe rozszerzeniom mogą być przeglądane przez najpopularniejsze przeglądarki internetowe (Netscape i MS IE). Można nawet tak skonfigurować niektóre serwery WWW aby dostarczały odbiorcy pojedyncze strony dokumentów wielostronicowych w trybie „na żądanie”.

Co najważniejsze, specyfikacja PDF została udostępniona przez firmę Adobe a program Acrobat Reader pozwalający na przeglądanie i drukowanie plików PDF został przygotowany na bardzo różne platformy (DOS, Windows, Macintosh, Unix) i jest udostępniany za darmo.

Ciągle trwają przymiarki stworzenia niewielkiego (format strony książki) komputera (e-Book), który wykorzystując technologie Adobe (lub podobną) pozwoli na skonstruowanie „wirtualnej” książki.

Generalnie, wydaje się, że stworzenie dokumentu PDF jest łatwe:

1. Za pomocą dowolnego programu (procesora tekstu) nadajemy słowom odpowiednią formę zewnętrzną.
2. Przekształcamy wszystko do postaci pliku PostScriptowego (na przykład „drukując” do pliku, z wykorzystaniem sterownika PostScript).
3. Otrzymany plik przetwarzamy („destylujemy”) do postaci PDF używając jakiegoś przeznaczonego do tego programu.

2. Programy do przetwarzania pliku PS do postaci PDF

Podstawowym narzędziem używanym do konwersji PS \rightarrow PDF jest Acrobat Distiller. Program dostępny jest na platformy MS Windows, Macintosh i Unix (tylko wersja 3 i nie jest dostępny dla Linuksa). Distiller produktem komercyjnym.

Innym programem, udostępnianym na licencji *shareware*, jest PStill¹. Wersja pracująca w środowisku Unix/Linux jest darmowa do zastosowań osobistych.

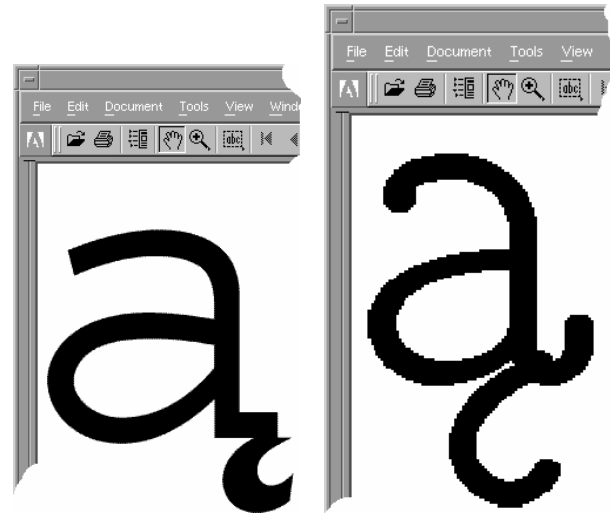
Kolejnym narzędziem rozpowszechnianym na jeszcze mniej ograniczających warunkach licencji jest Aladdin Ghostscript². Niestety, ostatnia stabilna, podczas pisania tych słów, wersja programu (5.50) nie może jeszcze dorównywać produktowi Adobe – ma między innymi kłopoty z przetwarzaniem fontów dołączonych do pliku ps [3]. Wedle wszelkich zapowiedzi wersja 6 będzie to już robiła poprawnie.

3. (La)TeX i PDF

Zgodnie z tym co napisałem wcześniej, naturalną metodą tworzenia dokumentów PDF z wykorzystaniem (La)TeX-a jest kompilacja do postaci DVI, a następnie przekształcenie pliku DVI do postaci PostScriptowej za pomocą, na przykład, programu dvips. Plik PostScriptowy można już poddawać „klasycznej” obróbce.

O ile jednak system przetwarzania tekstów TeX dostępny jest dla wszystkich za darmo o tyle dalszy ciąg przetwarzania musi być już dokonywany za pomocą oprogramowania komercyjnego, jeżeli tylko zależy nam na przyzwoitej jakości.

W chwili obecnej dostępne są w większości spotykanych dystrybucji systemu TeX dwa, a właściwie



Rysunek 1: Przykład dokumentu złożonego różnymi fontami, po lewej stronie font skalowalny, po prawej EC

nawet trzy, bardzo intensywnie rozwijane, narzędzia do tworzenia plików PDF w środowisku TeX-owym:

- dvi2pdfm, autor: Mark A. Wicks [5];
- pdfTeX, autor: Hàn Thê Thành [1].

Trzecim narzędziem (nie będzie dalej omawiane) jest komercyjny³ system VTeX firmy MicroPress⁴.

Trudno dziś powiedzieć który z programów jest lepszy: pierwszy korzysta ze standardowego kompilatora systemu TeX i poddaje interpretacji plik DVI. Drugi tworzy bezpośrednio w procesie kompilacji pliku źródłowego albo plik DVI albo plik PDF. Trzeci ma wiele bardzo interesujących możliwości, których jeszcze nie zdołałem rozpoznać...

Większość wspomnianych wyżej problemów, które możemy napotkać w procesie tworzenia dokumentów PDF jest wspólna dla wszystkich narzędzi.

4. Fonty

Na czym polega problem z fontami? Otóż Acrobat Reader potrafi bardzo ładnie wyświetlać na ekranie znaki tworzone przy pomocy czcionek skalowalnych (Type 1, TrueType). Jeżeli jednak mamy do czynienia z czcionką rastrową (Type 3) otrzymany rezultat jest niezbyt dobry (spodziewać się można efektów podobnych do tych pokazanych na rysunku 1) a strony wyświetlane są znacznie wolniej.

¹ <http://www.this.net/~frank/>.

² <http://www.aladdin.com>.

³ Wersja dla systemu Linux jest darmowa.

⁴ <http://www.micropress-inc.com>.

Problem nie dotyczy w tym samym stopniu drukowania: plik PS będący podstawą do tworzenia pliku PDF budowany jest najczęściej w rozdzielczości 300 lub 600 dpi i drukowany w zbliżonej rozdzielczości.

Aby osiągnąć właściwy efekt powinniśmy tak skonfigurować oprogramowanie aby używało odpowiednich fontów: Type 1 lub TrueType. Powinniśmy również zainstalować w swoim systemie te fonty i wszystkie potrzebne do pracy \TeX -a i \dvips -a pliki z fontami związane. Tak się składa, że zarówno fonty Computer Modern, jak i oparte na nich fonty PL dostępne są w formacie Type 1.

Następnie należy przygotować dokument źródłowy w taki sposób aby skład tekstu wykonywany był z wykorzystaniem tych fontów.

„Klasyczne” preambuły tworzonych w \LaTeX dokumentów wyglądają jak poniżej: najpierw z wykorzystaniem pakietu \PLATeX :

```
%& --translate-file=cp1250pl
\documentclass[ ]{}
\usepackage[OT4]{polski}
```

```
\begin{document}
```

```
...
```

```
\end{document}
```

albo Babel:

```
%& --translate-file=cp1250pl
\documentclass[ ]{}
\usepackage[polish]{babel}
\usepackage[OT4]{fontenc}
\begin{document}
```

```
...
```

```
\end{document}
```

Ostatni przykład pozwala na wykorzystanie standardowych fontów PostScript:

```
%& --translate-file=cp1250pl
\documentclass[ ]{}
\usepackage[T1]{polski}
\usepackage{pslatex}
\begin{document}
```

```
...
```

```
\end{document}
```

Zwracam uwagę, że próba rezygnacji z wykorzystania fontów PL czyli zamiana parametru pakietu fontenc z OT4 na T1 nie pozwala na uzyskanie zadowalającego efektu: zarówno \LaTeX jak i \pdfLaTeX używać będą fontów EC, które zawierają wszystkie znaki niezbędne dla tworzenia dokumentów w językach łacińskich, ale nie są dostępne w postaci

skalowalnej. Prawe okno programu Acrobat na rysunku 1 obrazuje właśnie ten przypadek. Z tego powodu można zalecić dodatkowo użycie pakietu pslatex (efekt widać na lewym oknie na rysunku 1).

Jeżeli chcemy tworzyć ładne pliki PDF w języku polskim, musimy mieć odpowiednie (to znaczy zawierające polskie litery) fonty Type 1 wraz ze wszystkimi niezbędnymi plikami do ich użycia w $(\text{La})\text{\TeX}$ -u. Oprócz fontów PL (kodowanie OT4) nie mamy zbyt wielkiego wyboru, jeżeli chodzi o fonty darmowe: QuasiTimes i QuasiPalladio oraz Antykwia Toruńska to jedyne znane mi fonty z polskimi literami i z „pełnym” wsparciem dla \TeX -a.

Możemy również skorzystać z „protezy”: użyć pakietu otlpatch rozpowszechnianego wraz z pakietem \PLATeX („dorabia” on ogonki do fontów CM) albo pakietu ae⁵ albo (i to wydaje się być rozwiązaniem najrozsądniejszym, choć nie idealnym) pakietu pslatex. Dodatkową zaletą użycia tego pakietu jest to, że wykorzystuje on kroje Times, Courier i Helvetica, a więc te, które są dołączane do programu Acrobat Reader – zatem nie muszą być włączane do dokumentu.

Wadą takiego rozwiązania jest to, że polskie litery mają bardzo niestarannie „przyklejone” ogonki, najbardziej drastyczny przykład poniżej:

Użycie programu Ghostscript do konwersji pliku PS do postaci PDF powoduje, że wszystkie „niestandardowe” (to znaczy dołączane w pliku PS) skalowalne fonty są przez niego konwertowane do postaci Type 3 (zamieniane do postaci rastrowej).

Zamiast fontów w formacie Type 1 można wykorzystać fonty TrueType (tylko \pdfTeX). Szczegółowy opis jak to zrobić znajduje się na stronie <http://quantum.bitp.kiev.ua/radamir/ttf-tex.htm>; autorem jest Damir Rakityansky.

Pojawia się tu jednak kolejny problem związany ze specyfiką formatu PDF: wszystkie niestandardowe fonty muszą być włączone do wynikowego dokumentu. Ponieważ fonty objęte są na świecie ochroną wynikającą z praw autorskich – trzeba z tym bardzo uważać jeżeli chcemy utworzony dokument komuś przekazać lub udostępnić w Internecie:

⁵ Każde archiwum lustrzane CTAN, np: <http://sunsite.icm.edu.pl/pub/CTAN/fonts/ae/>.

licencja na wykorzystanie zakupionych fontów może nie obejmować tego przypadku.

5. Grafika

Włączanie grafik do plików PDF jest (bardzo) proste: jeżeli korzystamy z \LaTeX -a powinniśmy użyć pakietu `graphicx` i postępować dokładnie tak samo, jak w przypadku klasycznym (patrz również [2]).

Jedyny problem, który tu powstaje, wynika z tego, że każdy z wymienionych programów (`dvips`, `pdfTeX` oraz `dvipdfm`) korzysta z innych formatów graficznych (lub innych instrukcji do włączania grafik). Opcjonalny parametr w wywołaniu pakietu `graphicx` podpowiada jakie konkretnie instrukcje mają być użyte do włączania plików graficznych.

```
\usepackage[ ]{graphicx}
```

Powoduje to pewne kłopoty przy tworzeniu plików źródłowych, które mają służyć do klasycznego przetwarzania ($\text{\TeX} \rightarrow \text{\DVI} \rightarrow \text{\PS}$), jak i tworzenia plików PDF w inny sposób.

Rozpatrzmy dwie sytuacje: przetwarzanie trój-etapeowe, z wykorzystaniem `dvips` i jednoetapeowe: `pdfTeX`. W pierwszym przypadku wszystkie pliki graficzne muszą być przekształcone do formatu EPS. W drugim, program akceptuje plik PDF, PNG i JPG. Załóżmy, że mamy trzy pliki graficzne: `a.jpg`, `b.png` i `c.eps`. Ten ostatni zawiera grafikę wektorową. Musimy przetworzyć go do postaci PDF.

Program `pdfTeX` nie pozwala na konwersję „w locie” plików graficznych. Aby zaoszczędzić miejsce na dysku, będziemy dokonywali „w locie” konwersji plików JPG i PNG do formatu EPS.

```
%& --translate-file=cp1250pl
\documentclass[ ]{ }
\newif\ifpdf
\ifx\pdfoutput\undefined \pdffalse
\else \pdfoutput=1 \pdftrue \fi
\usepackage[OT4]{polski}
\ifpdf \usepackage[pdftex]{graphicx}
\DeclareGraphicsExtensions{.pdf,.jpg,.png}
\else
\usepackage[dvips]{graphicx}
\DeclareGraphicsExtensions{.png,.eps,.jpg}
\DeclareGraphicsRule{.jpg}{eps}{.jpg.bb}%
  {'jpeg2ps -q -r0 -h #1}
\DeclareGraphicsRule{.png}{eps}{.png.bb}%
  {'convert #1 eps:-}
\fi

\begin{document}
```

```
\includegraphics[width=5cm]{a}
\includegraphics[width=5cm]{b}
\includegraphics[width=5cm]{c}
\end{document}
```

Pierwsze linie przykładu definiują nowe polecenie `\ifpdf` pozwalające rozróżnić obie sytuacje: przetwarzanie za pomocą programu \LaTeX i za pomocą programu `pdfTeX`. Dalej korzystamy z nowej instrukcji definiując dwa środowiska pracy i rozpoznawane w nich typy plików graficznych. W przypadku kompilacji trójetapeowej zostały również zdefiniowane reguły konwersji $\text{JPG} \rightarrow \text{EPS}$ i $\text{PNG} \rightarrow \text{EPS}$. Wykorzystujemy w tym celu programy, które potrafią wynik konwersji zapisać „na terminal”. Są to `jpeg2ps`⁶ i `convert` z pakietu `ImageMagick`⁷.

Zasadnicza część dokumentu jest już jednakoowa dla każdego przypadku przetwarzania. Wraz z plikiem źródłowym mamy dodatkowo pliki graficzne: `a.jpg`, `b.png`, `c.eps` i `c.pdf` oraz dwa pliki zawierające wymiary generowanych w locie grafik (musimy przygotować je sami, podpatrując wynik przetwarzania wykorzystanych programów) `a.jpg.bb` i `b.png.bb`, zawierające linię podobną do tej (patrz również [4]):

```
%\BoundingBox: 20 20 70 70
```

Podobnej sztuczki nie da się generalnie zastosować w przypadku użycia programu `dvipdfm`. Akceptuje on grafiki w formatach PDF, JPG, PNG. Posiada również „wbudowany” mechanizm konwersji „w locie” $\text{EPS} \rightarrow \text{PDF}$ (z wykorzystaniem programu `GhostScript`)⁸. Ponieważ jednak przetwarzanie odbywa się z wykorzystaniem programu $(\text{La})\text{\TeX}$, nie ma łatwej (to znaczy bez modyfikacji pliku źródłowego) metody rozróżnienia jak powinny być przygotowywane pliki DVI: dla programu `dvips` czy `dvipdfm`.

Zestaw nagłówków, który można zaproponować dla programu `dvipdfm`, może wyglądać tak:

```
\usepackage[dvipdfm]{graphicx}
\DeclareGraphicsExtensions{.png,.eps,.jpg,.pdf}
\DeclareGraphicsRule{.jpg}{eps}{.jpg.bb}{ }
\DeclareGraphicsRule{.png}{eps}{.png.bb}{ }
\DeclareGraphicsRule{.pdf}{eps}{.pdf.bb}{ }
```

Rozpoznawane będą pliki typów (o rozszerzeniach) `.png`, `.eps`, `.jpg`, `.pdf`. Pliki `.png`, `.jpg` i `.pdf`

⁶ <http://www.muc.de/~tm/free/free.html>

⁷ <ftp://ftp.wizards.dupont.com/pub/ImageMagick/>

⁸ Nie zawsze, zwłaszcza w przypadku plików EPS o charakterze „rastrowym”, daje to dobre rezultaty!

nie będą poddawane konwersjom, ale musimy zadbać o przygotowanie odpowiednich plików pomocniczych zawierających informację o rozmiarach wstawianej grafiki. Dostarczany wraz z `dvipdfm` program `ebb` służy do tworzenia odpowiednich plików `.bb` na podstawie plików `PDF`, `PBG` i `JPG`⁹.

W poniższej tabeli zestawiono formaty plików graficznych akceptowanych przez pakiety $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -a:

	EPS	PS	PDF	MP	PNG	JPG
<code>dvips</code>	⊕	⊕	⊖	⊕	⊖	⊖
<code>pdfT_EX</code>	⊖	⊖	⊕	⊕	⊕	⊕
<code>dvipdfm</code>	⊙	⊖	⊕	⊕	⊕	⊕
<code>V_TE_X</code>	⊕	⊕	⊕	?	?	?

Uwagi: ⊕ – tak, ⊖ – nie, ⊙ – częściowo.

`dvips` – ogólne określenie wszystkich tych sposobów, które wykorzystują PostScript jako plik źródłowy do tworzenia pliku `PDF`.

`PS` – możliwość interpretacji poleceń języka PostScript (na przykład związanych z użyciem pakietu `PSTricks`).

Wnioski wypływające z tabeli są generalnie takie: pliki `EPS`, będące swojego rodzaju standardem grafik w środowisku $\text{T}_{\text{E}}\text{X}$ -owym, nie nadają się do bezpośredniego włączania do plików `PDF` (jeżeli nie chcemy korzystać z metody: `PS` → `PDF`). Podobnie, ani `pdfTEX` ani `dvipdfm` nie będą potrafiły zinterpretować praktycznie żadnych poleceń języka PostScript włączonych za pomocą `\special` (`PSTricks`, `psfrag`).

Jedynie plik `EPS` tworzone przez program `METAPOST` mają na tyle prostą strukturę, że zarówno `pdfTEX` jak i `dvipdfm` potrafią przetłumaczyć je na elementarne polecenia `PDF`.

Jak zatem przygotowywać grafiki, które chcemy włączać do dokumentów `PDF`:

grafiki wektorowe powinny być przekształcone, o ile to tylko możliwe, do postaci `PDF`; programami, które można wykorzystać w tym celu są: `distiller`, `epstopdf` i `PStill`. Najlepszy jest ten pierwszy. `epstopdf` jest to prosty program w Perlu wykorzystujący `Aladdin Ghostscript` do interpretacji PostScriptu i dodający do pliku informację o „`BoundingBox`”. Program obarczony jest wszystkimi wadami wspomnianymi wcześniej (fonty!). `PStill` poprawnie radzi sobie z zadaniem, chociaż tworzone pliki `PDF` są zazwyczaj duże i program zna tylko PostScript Level 1.

grafiki rastrowe. Tu powinniśmy użyć formatu `PNG` w przypadku grafik, w których zależy nam

⁹ Generowane przez `ebb` pliki `.bb` nie mogą być wykorzystane przez program `dvips`.

na pokazaniu szczegółów (skanowane rysunki techniczne, itp) lub formatu `JPG` w przypadku grafik z półtonami (zdjęcia). Format `JPG` ma ogromną zaletę: utworzone pliki wynikowe są stosunkowo małe. Ale ten efekt osiągany jest dzięki kompresji ze stratą jakości. Większość dobrych programów graficznych ma możliwość zapisywania w obu wspomnianych formatach. Oprócz tego, w pakiecie programów `netpbm`¹⁰ są narzędzia, które nieco „okrężną” drogą pozwalają na przekształcenie niemal dowolnego formatu graficznego w inny. Kolejnym darmowym narzędziem jest program `ImageMagick`.

W przypadku włączania grafik rastrowych pamięta trzeba, że program `distiller` pozwala na bardzo zaawansowane przetwarzanie grafik rastrowych. Może on stosować różne techniki określane mianem „`resamplingu`” (generalnie zmiana rozdzielczości włączanej grafiki) oraz kompresji (łącznie ze stratną kompresją typu `JPEG`). `dvipdfm` i `pdfTEX` tylko włączają gotową mapę bitową; skalowania i obroty wykonywane są podczas interpretacji pliku `PDF`.

Bibliografia

- [1] Hàn Thé Thành, Sebastian Rahtz, Hans Hagen. *The pdfT_EX manual*. <http://www.pragma-ade.nl/pdfTEX.htm>. Wrzesień 1999.
- [2] Wojciech Myszka. *Włączanie grafik w formacie EPS do tekstów w L^AT_EX 2_ε i parę innych uwag*. Dokument dostępny w postaci elektronicznej: <http://www.immt.pwr.wroc.pl/~myszka/grafika/>. Ciągłe uzupełniany. Październik 1999.
- [3] Tomasz Przechlewski. *Publikacje elektroniczne: T_EX i PDF. Biuletyn Polskiej Grupy Użytkowników Systemu T_EX*, Zeszyt 10, 1997. Tekst dostępny w <http://www.gust.org.pl/>.
- [4] Keith Reckdahl. *Using Imported Graphics in L^AT_EX 2_ε*. Znakomity tekst poświęcony włączaniu grafik `EPS` do tekstów w $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ 2_ε. Dostępny w postaci elektronicznej jako: <http://sunsite.icm.edu.pl/pub/CTAN/info/epsLATEX.pdf>. Grudzień 1997.
- [5] Mark A. Wicks. *Dvipdfm user's manual*. <http://odo.kettering.edu/dvipdfm/>. Wrzesień 1999.

◇ Wojciech Myszka
W.Myszka@immt.pwr.wroc.pl

¹⁰ Znajduje się w dystrybucji `MikTEX`-a; dostępny jest też na większość platform Unixowych.