



KATEDRA INFORMATIKY
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY
UNIVERZITA KOMENSKÉHO, BRATISLAVA

POSTSCRIPT PODPORA PRE PDF_TEX
(bakalárska práca)

PETER GATIAL

Odbor: Informatika 9.2.1
Vedúci: Mgr. Vladimír Koutný

Bratislava, 2007

Abstrakt

Gatial Peter, PostScript podpora pre pdfT_EX. Bakalárska práca, Katedra Informatiky, Fakulta Matematiky, Fyziky a Informatiky, Univerzita Komenského, Bratislava, vedúci bakalárskej práce: Mgr. Vladimír Koutný, Bratislava 2007.

PostScript je programovací jazyk, ktorý sa používa na popis vzhľadu strany. pdfT_EX je sázací systém určený na písanie hlavne matematických kníh, ktorého výstupom je PDF. V práci sa zaoberáme možnosťami ich vzájomnej spolupráce, kde PostScript využívame na kreslenie grafiky.

Práca vysvetľuje ako program T_EX funguje a ako používať PostScript v pdfT_EXu. Prináša prehľad o dostupných riešeniach, balíkoch pdfTricks a ps4pdf, resp. pst-pdf a navrhuje niekoľko vylepšení. Zaoberá sa aj ich použitím v praxi.

Kľúčové slová: PostScript, PDF, pdfT_EX, pdfTricks, PSTricks.

Abstract

Gatíal Peter, PostScript support for pdf \TeX . Bachelor thesis, Department of Computer Science, Faculty of Mathematics, Physics and Informatics, Comenius University, Bratislava, thesis advisor: Mgr. Vladimír Koutný, Bratislava 2007.

PostScript is programming language which is used to describe appearance of page. pdf \TeX is typographical system designed for writing especially mathematical books with output to PDF. In this paper we are dealing with possibilities of their co-operation where PostScript is used for drawing graphics.

This paper explains how the \TeX programme works and how to use PostScript in pdf \TeX . It presents list of available solutions, packages pdfTricks and ps4pdf, resp. pst-pdf and suggests some improvements. It also deals with their practical usage.

Key words: PostScript, PDF, pdf \TeX , pdfTricks, PSTricks.

Čestne prehlasujem, že som túto bakalársku prácu
vypracoval samostatne s použitím citovaných zdro-
jov.

.....

Ďakujem svojmu vedúcemu bakalárskej práce Mgr. Vladimírovi Koutnému za zadanie zaujímavej témy a za cenné námety pri jej spracovaní.

Obsah

1	Úvod	1
2	Formáty elektronických dokumentov	3
2.1	PostScript	3
2.2	Portable Document Format	4
2.2.1	Vlastnosti PDF	5
2.3	pdfTeX	6
3	pdfTeX a PostScriptové makrá	7
3.1	Ako funguje TeX	7
3.2	Formulácia problému	8
3.3	Riešenie	9
3.4	PSTricks	10
3.4.1	Používanie	11
3.4.2	Rozšírenia PSTricks	11
4	pdfTricks a ps4pdf	13
4.1	pdfTricks	13
4.1.1	Používanie	14
4.1.2	Návrhy na vylepšenia	15
4.2	ps4pdf	18
4.2.1	Používanie	19
4.3	pst-pdf	19
5	Použitie pdfTricks a ps4pdf v praxi	21
5.1	GasTeX	21
5.2	pst-3dplot	23
5.3	Beamer	23

<i>OBSAH</i>	ix
6 Záver	24
A Obsah priloženého CD	26
Literatúra	28

Zoznam obrázkov

3.1	Proces tvorby čitateľného výstupu pomocou $\text{T}_{\text{E}}\text{X}$ u	8
3.2	Manuálne vkladanie obrázkov do výsledného dokumentu	10
3.3	Jednoduchý príklad používania PSTricks	11
3.4	Príklad balíka pst-eucl - opísaná kružnica trojuholníka ABC	12
4.1	Postup práce pdfTricks	14
4.2	Porovnanie dokumentov pre PSTricks a pdfTricks	16
4.3	Porovnanie dokumentov pre PSTricks a ps4pdf	20
5.1	Gas $\text{T}_{\text{E}}\text{X}$ ový dokument pre pdfTricks a ps4pdf	22
5.2	Zodpovedajúci výstup k obrázku 5.1 pre pdfTricks	22
5.3	Ukážka pst-3dplot s pdfTricks	23

Zoznam tabuliek

A.1	Adresár /bin.	26
A.2	Adresár /doc – citovaná literatúra.	26
A.3	Adresár /example – príklady odkazované v práci.	27
A.4	Adresár /source – zdrojový kód práce.	27
A.5	Adresár /zmeny – súbory s navrhovanými zmenami.	27

Kapitola 1

Úvod

\TeX je veľmi známy sádzací program, ktorý naprogramoval Prof. Donald Ervin Knuth ešte v sedemdesiatych rokoch minulého storočia. Je určený na písanie kníh, ktoré obsahujú hlavne veľa matematikých vzorcov. Výstup \TeX u je vo formáte DVI. Ten však nie je pre človeka čitateľný, pretože môže obsahovať špeciálne príkazy, ktoré ešte musia byť spracované nejakým DVI prekladačom na čitateľný výstup.

Ako sádzací systém však \TeX nie je primárne určený na tvorbu grafiky. Preto ak chceme mať v nejakej knihe napríklad graf funkcie, musíme si ho najskôr nakresliť pomocou iného programu a vložiť do \TeX u ako obrázok. Iný spôsob je priamo v \TeX u použiť PostScript. \TeX totiž takýto PostScriptový kód len zapíše do DVI výstupu, kde môže byť spracovaný prekladačom, ktorý PostScriptu rozumie.

V poslednej dobe sa však výrazne rozšíril formát PDF. Oproti PostScriptu ponúka mnohé výhody a to hlavne v elektronickej podobe. Preto je čoraz obľúbenejší aj na písanie odborných prác z oblasti matematiky, fyziky alebo informatiky. Hoci existujú mnohé nástroje na konverziu medzi PostScriptom a PDF, takýto postup nie je vhodný, pretože PostScript nemá podporu pre niektoré interaktívne prvky PDF. Preto potrebujeme vyrobiť PDF priamo a nie konvertovaním z PostScriptu.

Keďže \TeX sa neustále rozvíja, jeho bežnou súčasťou je dnes aj rozširujúci program $\text{pdf}\text{\TeX}$, ktorého výstupom je priamo PDF. Nevýhodou však je, že nerozumie PostScriptovému kódu, a preto PostScriptové makrá nemôžeme v $\text{pdf}\text{\TeX}$ u štandardným spôsobom použiť.

Hlavným cieľom tejto práce je preto vysvetlenie ako používať PostScriptové makrá v $\text{pdf}\text{\TeX}$ u. Čitateľ by mal rozumieť pojmom ako sú vstupný a

výstupný súbor a ako vytvoriť textový súbor. Neočakáva sa však, že čitateľ je expert na prácu s $\text{T}_{\text{E}}\text{X}$ om, hoci aspoň základná predstava je vítaná. V práci sa nebudeme venovať inštalácii $\text{T}_{\text{E}}\text{X}$ u ani spomínaných balíkov.

Štruktúra bakalárskej práce

Kapitola 2 obsahuje popis oboch spomínaných formátov elektronických dokumentov, ktorým sa budeme v tejto práci venovať. Uvedieme tu ich vlastnosti a výhody. Záver tejto kapitoly patrí programu pdf $\text{T}_{\text{E}}\text{X}$.

Kapitola 3 je jednou z dvoch hlavných kapitol tejto bakalárskej práce. Venuje sa problematike použitia PostScriptových makier v pdf $\text{T}_{\text{E}}\text{X}$ u. V časti 3.1 najskôr oboznámime čitateľa s tým ako funguje $\text{T}_{\text{E}}\text{X}$, aby sme v časti 3.2 mohli zanalyzovať a vysvetliť, prečo nie je možné štandardným spôsobom používať PostScriptové makrá v pdf $\text{T}_{\text{E}}\text{X}$ u. V nasledujúcej časti rozoberieme niekoľko spôsobov ako sa dá tento problém riešiť. V závere tejto kapitoly sa budeme zaoberať najznámejším balíkom PostScriptových makier.

Kapitola 4 obsahuje popis dvoch balíkov, ktoré umožňujú použitie PostScriptových makier v pdf $\text{T}_{\text{E}}\text{X}$ u. Oba využívajú iný princíp a je na čitateľovi, aby si vybral, ktorý z týchto postupov mu viac vyhovuje.

Kapitola 5 ukazuje použitie takýchto balíkov v praxi.

Kapitola 2

Formáty elektronických dokumentov

V tejto kapitole si popíšeme dva, v matematickej komunite veľmi rozšírené, formáty elektronických dokumentov, ktorými PostScript a PDF nepochybne sú. Stručne zhrnieme ich vlastnosti a rozdiely, ktoré sú dôležité pre pochopenie, prečo sa zaoberáme problematikou použitia PostScriptových makier v pdf \TeX u. V závere kapitoly stručne popíšeme aj samotný program pdf \TeX .

Pod pojmom *TEXový dokument* budeme rozumieť syntakticky korektný vstupný súbor pre \TeX , ktorý môžeme vytvoriť v ľubovoľnom textovom editore. *PDF dokument* bude výstupom programu pdf \TeX . Pokiaľ bude z kontextu vyplývať, o akom type dokumentu hovoríme, budeme niekedy slovo *TEXový*, resp. *PDF* vynechávať.

2.1 PostScript

PostScript [Inc99] je známy programovací jazyk vytvorený firmou Adobe Systems Incorporated. Na trh bol uvedený v roku 1985 a od tej doby bol už niekoľkokrát rozšírený. Je určený na popis textu, grafických útvarov a obrázkov zobrazených na stranách, pričom tento popis je nezávislý na výstupnom zariadení a rozlíšení. Popisovacie schopnosti tohoto jazyka zahŕňajú nasledovné možnosti, ktoré môžu byť ľubovoľne kombinované:

- Ľubovoľné grafické útvary skladajúce sa z priamych čiar, oblúkov, obdĺžnikov a kriviek, ktoré sa môžu navzájom prekrývať.

- Operátory, ktoré umožňujú, aby jednotlivé útvary boli nakreslené čiarami ľubovoľnej hrúbky, vyplnené ľubovoľnou farbou alebo použité na orezávanie inej grafiky. Farby môžu byť zadané v rôznych formátoch ako sú RGB, CMYK a iné.
- Plne integrovaný text s grafikou. V skutočnosti sú jednotlivé znaky vo fontoch chápané ako grafické útvary, s ktorými môžeme pracovať pomocou grafických operátorov tak isto ako s akoukoľvek inou grafikou, a tak text vypísať v ľubovoľnom rozlíšení.
- Podpora lineárnych transformácií ako sú posunutie, škálovanie alebo rotácia. Tieto transformácie sa aplikujú rovnako na všetky prvky na strane, vrátane textu, grafických útvarov a obrázkov.

O PostScripte môžeme naozaj hovoriť ako o programovacom jazyku, pretože má tradičné dátové typy ako sú čísla, polia a znakové reťazce; základné riadiace štruktúry ako podmienky, cykly a procedúry, ale aj niektoré neobvyklé vlastnosti ako napríklad slovníky. Toto všetko umožňuje programátorom veľmi presne popísať výsledný vzhľad strany a nakresliť tak náročnú a detailnú grafiku. Preto je PostScript veľmi populárny pri písaní T_EXových balíkov (angl. package).

Podrobnejší popis fungovania PostScriptu a písania PostScriptových programov presahuje rámec tejto práce, v prípade potreby ho čitateľ nájde napr. v knihách [Inc85] alebo [Rei90].

2.2 Portable Document Format

Portable Document Format [Inc06], dnes známy pod skratkou PDF, je tiež formát vytvorený firmou Adobe Systems Incorporated. Keď sa tento formát v roku 1993 objavil, nepodarilo sa mu presadiť okamžite. Dôvodom bolo to, že nástroje na tvorbu, ale hlavne aj prehliadače, neboli zadarmo. Ďalej, prvé verzie nemali podporu pre externé hyperlinky, čo obmedzovalo ich využitie na internete a postavenie PDF ohrozovali aj konkurenčné formáty iných firiem a dokonca aj samotný PostScript. Od svojho vzniku však bolo PDF už niekoľkokrát aktualizované a postupne sa stalo štandardom v oblasti elektronických dokumentov.

2.2.1 Vlastnosti PDF

PDF je nariadením od PostScriptu formát a nie programovací jazyk, a preto nemá napríklad procedúry alebo cykly, ktoré umožňujú PostScriptu vykonávať častokrát veľmi dôležité výpočty. Tento formát bol navrhnutý tak, aby splnil niekoľko cieľov.

- Umožniť užívateľom výmenu a prezeranie elektronických dokumentov ľahko, spoľahlivo a nezávisle na prostredí, v ktorom boli vytvorené.
- Popis textu a grafiky je nezávislý na zariadení a rozlíšení.
- Dokumenty sú optimalizované pre prezeranie. Pre zlepšenie interaktívneho prezerania definuje PDF viac štruktúrovaný formát ako PostScript. PDF obsahuje objekty ako sú anotácie a hypertextové odkazy, ktoré nie sú súčasťou strany, ale sú užitočné pre interaktívne prezeranie a výmenu dokumentov.

Je zrejmé, že PDF ako nový formát má oproti PostScriptu niekoľko výhod, keďže disponuje modernými vlastnosťami ako sú:

- **Dostupnosť** - dokumenty môžu byť prispôbené ľuďom s rôznymi postihnutiami. PDF môže obsahovať tagy, prepisy textov alebo zvukové nahrávky, ktoré môžu byť nahlas čítané špeciálnymi softvérmi. Pre zle vidiacich ľudí tu je možnosť zobrazíť si dokument v ľubovoľnom rozlíšení.
- **Font manažment** - všetky fonty použité v PDF môžu byť pripojené k súboru, a tak text bude vyzerať vždy rovnako.
- **Kompresia** - PDF objekty, hlavne obrázky, môžu byť skompresované rôznymi algoritmami, a tak sa celková veľkosť PDF súboru môže výrazne zmenšiť, čo je výhodné pre webové prezeranie.
- **Bezpečnosť** (šifrovanie, digitálny podpis) - PDF môže byť zašifrované tak, že na prezeranie alebo editovanie je potrebné heslo.
- **Priamy prístup na stranu** - PostScript je imperatívny programový jazyk s globálnym stavom, takže inštrukcie pre popis jednej strany môžu ovplyvniť nasledujúce. Preto skôr než môže byť nejaká strana zobrazená, musia byť najskôr spracované všetky predchádzajúce strany.

V PDF je každá strana nezávislá a nie je nijako ovplyvnená ostatnými stranami. Toto umožňuje rýchle načítanie ktorejkoľvek strany, čo je výhodné pre webové prezeranie.

2.3 pdfTeX

Keďže výstupom TeX-u sú DVI¹ súbory, dlhú dobu nebolo možné vytvárať PDF z TeXového dokumentu priamo. Jeden zo spôsobov, ako sa to mohlo realizovať, pozostával z niekoľkých krokov:

1. skontrolovať TeXový súbor na DVI súbor, ktorý obsahuje `\special` príkazy potrebné pre PDF
2. skontrolovať DVI súbor na PostScriptový súbor napríklad známym programom `\dvips`²
3. a nakoniec skontrolovať PostScriptový súbor na PDF napríklad pomocou programu Ghostscript³.

Ďalšia možnosť bola použiť program `dvipdf` alebo `dvipdfm`, ktoré celý tento proces uľahčovali tým, že vynechali PostScriptový medzikrok. Tieto programy však neboli úplne dokonalé a nepodporovali niektoré vlastnosti.

Zmena nastala až v roku 1998 kedy začala práca na programe pdfTeX. Dnes je pdfTeX azda najrozšírenejší program na tvorbu PDF dokumentov v TeXovej komunite a je štandardnou súčasťou najznámejších TeXových distribúcií ako sú TeX Live alebo MikTeX.

V skutočnosti je to rozšírená verzia TeX-u, ktorú naprogramoval Hàn Thê Thành ako postgraduálny projekt. Hlavným rozdielom medzi TeX-om a pdfTeX-om je to, že zatiaľ čo výstupom TeX-u sú DVI súbory, pdfTeX dokáže vytvárať PDF priamo z TeXových súborov bez generovania DVI. To umožňuje využiť vlastnosti, ktoré ponúka PDF a vylepšiť tak výsledok sádzania TeX-u. Na druhej strane, balíky ako PSTricks, ktoré na svoju prácu používajú PostScript, nemusia fungovať.

Keďže L^ATeX je len balík makier pre TeX, pracuje rovnako aj s pdfTeX-om. Preto pdfL^ATeX volá pdfTeX pričom používa štandardné L^ATeXové makrá na vysádzanie L^ATeXového dokumentu.

¹odvodené z Device Independent

²voľne dostupný DVI prekladač

³voľne dostupný PostScriptový interpretér

Kapitola 3

pdfT_EX a PostScriptové makrá

V tejto kapitole sa pokúsime formulovať problematiku použitia PostScriptových makier v pdfT_EXu. Najskôr si stručne vysvetlíme ako funguje T_EX a jeden z jeho príkazov. Ďalej popíšeme proces, ktorým z T_EXového dokumentu vzniká PDF dokument a kde v tomto procese nastáva problém. Nakoniec sa pozrieme na najznámejší balík PostScriptových makier v T_EXu a v krátkosti si popíšeme jeho použitie.

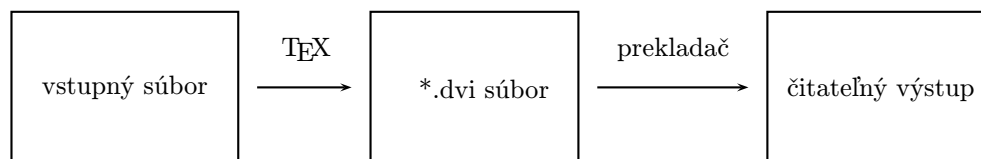
3.1 Ako funguje T_EX

Ako už bolo spomenuté, vstupom pre program T_EX sú textové súbory. Ich obsahom je vlastný text dokumentu, ktorý po spracovaní vidíme, ale aj príkazy začínajúce sa špeciálnym znakom `\` ako napríklad `\includegraphics`. Tieto príkazy, ktorými usmerňujeme činnosť T_EXu, majú rôznu funkciu, od pomoci pri písaní a formátovaní výsledného textu až po vkladanie externých súborov.

Medzi všetkými príkazmi sa však nachádza jeden, ktorý má zvláštnu úlohu. Niekedy totiž potrebujeme vykonať veci, ktoré vie spraviť len DVI prekladač ale nie T_EX, a teda potrebujeme z T_EXového dokumentu preniesť nejakú informáciu priamo do výstupného DVI súboru. Napríklad T_EX nevie nič o tom ako nastaviť farbu textu, ale niektoré prekladače to dokážu. Pre tento prípad T_EX ponúka príkaz

$$\backslash\text{special}\{\text{text}\},$$

ktorého argumenty závisia na prekladači, pre ktorý je určený. Všetko čo T_EX

Obr. 3.1: Proces tvorby čitateľného výstupu pomocou TeX u

s týmto príkazom urobí, je expandovanie jeho argumentov a následne ho zapíše do DVI súboru. Takže pri spracovaní vstupného súboru môžeme pomocou tohoto príkazu docieľiť, aby TeX argument tohoto príkazu takpovediac ignoroval a len ho zapísal do DVI výstupu a nechal na spracovanie nejakému DVI prekladaču. Celý proces si teda môžeme predstaviť ako postup na obrázku 3.1.

Našťastie vo väčšine prípadov TeX ový užívateľ nemusí poznať a písať komplikované `\special` príkazy sám, pretože má k dispozícii rôzne balíky makier, ktoré tieto príkazy skrývajú. Štandardnou súčasťou $\text{L}^{\text{A}}\text{TeX}$ u sú napríklad balíky `graphics` alebo `color`, ktoré výrazne uľahčujú prácu s vkladaním, otáčaním a škálovaním obrázkov alebo zmenou farby textu. V prípade ďalších otázok môže čitateľ nájsť veľmi podrobný popis TeX u v [Knu84] alebo stručnejšie v [Doo87].

3.2 Formulácia problému

Keďže TeX bol naprogramovaný hlavne ako sádzací systém, nemá priamo podporu pre grafiku. Súčasťou TeX u je síce program `METAFONT`¹, ale ten je určený na dizajn a prácu s fontami a pre naše účely (kreslenie grafov funkcií alebo matematické výpočty) je príliš neohrabaný. $\text{L}^{\text{A}}\text{TeX}$ zasa ponúka užívateľsky jednoduché prostredie `\picture`, ktoré je však až príliš obmedzujúce a neposkytuje také prostriedky ako PostScript.

Preto sa na kreslenie grafiky používa priamo PostScript. Výstupom TeX u je však DVI, a tak sa musí všetka grafika využívajúca PostScript nachádzať v príkazoch `\special`, aby neskôr mohla byť vybratá a spracovaná DVI prekladačom ako je napríklad `dvips` na PostScriptový kód. Balíky ako `PSTricks` (a jeho rozšírenia) alebo `PSfrag` takéto `\special` príkazy vytvárajú. `pdfTeX`,

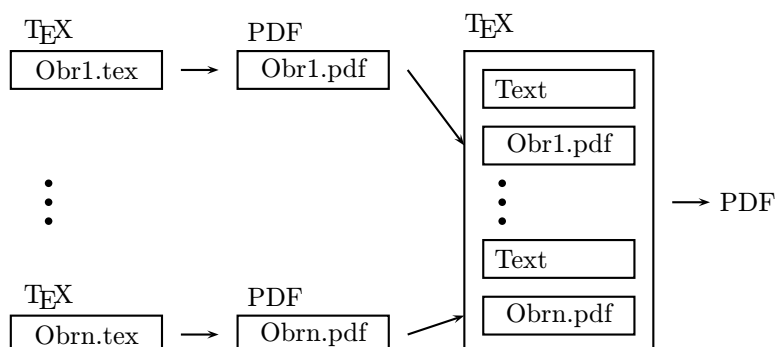
¹programovací jazyk tiež vytvorený Knuthom

ktorý však produkuje PDF priamo bez DVI, PostScriptovému kódu nerozumieme, a preto všetky `\special` príkazy, ktoré nie sú určené pre PDF, ignoruje.

3.3 Riešenie

S rozšírením formátu PDF sa ľudia prirodzene začali zaujímať o to ako by bolo možné použiť PostScriptový kód v PDF a využiť tak výhody oboch formátov. Cieľom bolo, aby na jednej strane bol celý proces užívateľsky maximálne nenáročný a na druhej, aby bolo možné využiť prezenčné vlastnosti PDF zároveň so silou PostScriptu.

1. Ako sme už spomenuli skôr, jedna možnosť je skonvertovať PostScriptový súbor na PDF. Celý postup by vyzeral takto: $\text{T}_{\text{E}}\text{X} \rightarrow \text{DVI} \rightarrow \text{PS} \rightarrow \text{PDF}$, teda presne ako na obrázku 3.1 len s tým, že ako DVI prekladač by sme použili `dvips` a potom ešte program Ghostscript. Pritom by sme ale stratili výhody PDF, keďže PostScript nepodporuje niektoré moderné vlastnosti PDF.
2. Ďalšia možnosť je najskôr všetku grafiku, ktorá používa PostScript, dať do samostatných súborov, z nich klasickým spôsobom vyrobiť PostScriptové súbory, tie už známym postupom skonvertovať na PDF a následne vložiť do dokumentu na želané miesto pomocou príkazov ako `\includegraphics`. Tento postup znázorňuje obrázok 3.2. Toto však vyžaduje pomerne veľa manuálnej práce alebo písanie skriptov, ktoré by to vykonali za nás. Ďalšou nevýhodou je, že by sme zbytočne mali množstvo malých súborov, čo nemusí byť vždy vhodné a žiadané. Preto ľudia hľadali nejaké univerzálnejšie riešenie a spôsob ako by sa dali niektoré časti tohoto postupu čo možno najviac zautomatizovať.
3. Zvoľme teraz iný prístup. Pozrime sa najskôr na to ako taký $\text{T}_{\text{E}}\text{X}$ ový dokument vyzerá. V zjednodušenej predstave môžeme povedať, že každý dokument je istým spôsobom štruktúrovaný na blok textu, obrázok, blok textu, atď. Ďalej, väčšina dokumentov je dnes písaná v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u, ktorý poskytuje na zjednodušenie práce prostredia (angl. environment), ktoré dokument štruktúrujú ešte výraznejšie. Preto by bolo výhodné nájsť spôsob, ktorým označíme časti dokumentu, ktoré obsahujú PostScriptový kód, nejako ich samostatne spracujeme, skonvertujeme na PDF a neskôr vložíme späť na pôvodné miesto do nášho dokumentu



Obr. 3.2: Manuálne vkladanie obrázkov do výsledného dokumentu

už v PDF formáte. Takto budeme môcť plne využiť vlastnosti PDF, pretože z pohľadu T_EXu budeme do výsledného PDF dokumentu len vkladať nejakú grafiku z externých súborov a teda môžeme bez ťažkostí pustiť pdfT_EX.

3.4 PStricks

Ako už bolo spomenuté, PostScriptový kód sa musí v T_EXovom dokumente nachádzať v príkaze `\special`. Z pohľadu užívateľa je však písanie takýchto príkazov nepohodlné a veľmi náročné, keďže je nutné poznať presný formát týchto príkazov, aby im DVI prekladač rozumel. Preto vznikli rôzne balíky, ktoré poskytujú jednoduché a intuitívne príkazy, ktoré tieto `\special` príkazy skrývajú, a tak uľahčujú prácu.

Jedným z takýchto balíkov je aj PStricks, ktorý vznikol pre jednoduchšie používanie PostScriptu v T_EXových dokumentoch. Je to balík, ktorý umožňuje využiť väčšinu PostScriptových vlastností. Môžeme ho použiť na kreslenie čiar, kriviek, polygónov alebo grafov funkcií.

PStricks má za sebou už dosť dlhý vývoj. Celý projekt začal Prof. Timothy Van Zandt. Odvtedy bol už výrazne prepracovaný a zdokonalený. V poslednom čase PStricks udržujú a ďalej rozvíjajú hlavne Denis Girou, Sebastian Rahtz a Herbert Voß. Dnes je to zrejme najznámejší balík pre prácu s PostScriptovými makrami v T_EXu a je súčasťou najznámejších distribúcií ako sú T_EX Live a MikT_EX.

3.4.1 Používanie

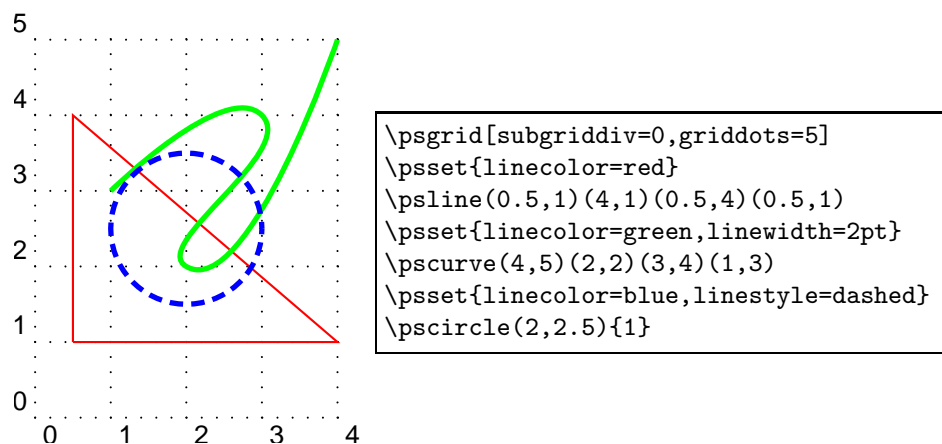
Práca s týmto balíkom je veľmi jednoduchá. Na začiatku dokumentu načítame balík pomocou príkazu

```
\usepackage{pstricks}.
```

PostScriptové makrá stačí napísať do `\pspicture` prostredia, ktoré je lepšie než `\picture` v L^AT_EXu. Základné použitie ukazuje obrázok²3.3. Na nastavovanie rôznych grafických parametrov ako sú farba výplne alebo hrúbka čiary slúži príkaz

```
\psset{par1 = value1, par2 = value2, ...}.
```

Skúsenejší užívatelia môžu písať aj priamo PostScriptový kód a to pomocou príkazu `\code`. Zložitejšie príklady čitateľ nájde na <http://www.tug.org/PSTricks/main.cgi?file=examples> a podrobný zoznam ďalších príkazov v dokumentácii [Zan03].



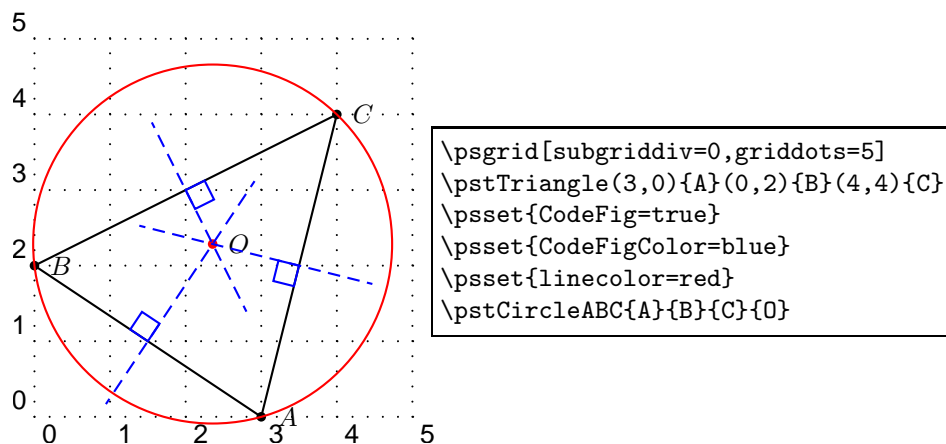
Obr. 3.3: Jednoduchý príklad používania PSTricks

3.4.2 Rozšírenia PSTricks

Počas svojej dlhej existencie si tento balík získal mnohých priaznivcov a časom boli samozrejme napísané aj mnohé rozšírenia, ktoré sa bližšie špecializujú len na vybrané oblasti. Napríklad `pst-plot` na kreslenie grafov funkcií,

²upravené a prevzaté z <http://en.wikipedia.org/wiki/PSTricks>

`pst-3dplot` na kreslenie 3-rozmerných grafov funkcií, `pst-circ` na kreslenie elektrických obvodov alebo `pst-eucl` na kreslenie geometrie. Príklad posledne uvedeného môžeme vidieť na obrázku³3.4. Podrobný zoznam môže čitateľ nájsť na <http://tug.org/PSTricks/main.cgi?file=packages>.



Obr. 3.4: Príklad balíka `pst-eucl` - opísaná kružnica trojuholníka ABC

³upravené a prevzaté z [Rod05]

Kapitola 4

pdfTricks a ps4pdf

V tejto kapitole sa bližšie pozrieme na balíky, ktoré umožňujú použitie PostScriptových makier v pdf \TeX u. Stručne si vysvetlíme ako pracujú a navrhne niekoľko vylepšení.

4.1 pdfTricks

Tento balík vznikol ako odpoveď na veľmi populárny balík PSTricks, aby umožnil použitie týchto makier aj v pdf \TeX u. Autormi sú C. V. Radhakrishnan a C. V. Rajagopal.

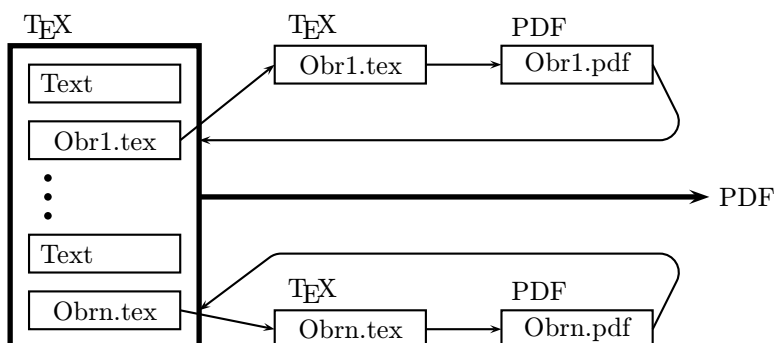
Balík funguje na princípe popísanom v časti 3.3. Pri spracovaní nášho dokumentu potrebujeme najskôr samostatne skonvertovať PostScriptové časti na PDF a následne ich vložiť späť na pôvodné miesto.

pdfTricks to zabezpečí pomocou funkcie "shell escape", ktorá je súčasťou web2c kompilátora \TeX u. Táto funkcia nám umožní zastaviť prácu \TeX u, vykonať ľubovoľný príkaz operačného systému a pokračovať ďalej. Z \TeX ového dokumentu dá vyvolať pomocou príkazu

```
\write18{príkaz}.
```

Tu je vhodné podotknúť, že "shell escape" funkcia je štandardne vypnutá a to z čisto bezpečnostných dôvodov, keďže dovoľuje vykonať ľubovoľný príkaz ako napríklad zmazanie súborov. Preto je nutné túto funkciu pri každom spustení \TeX u povoliť pomocou parametru `-shell-escape` takto:

```
pdflatex -shell-escape vstupny_subor.
```

Obr. 4.1: Postup práce pdfTricks

4.1.1 Používanie

Balík načítame známym príkazom

```
\usepackage{pdftricks}.
```

K dispozícii je aj niekoľko užitočných nastavení ako `debug` a `NoProcess`, ich popis čitateľ nájde v [RR03]. Na označenie PostScriptových častí dokumentu použijeme prostredie `\pdfpic` alebo `\pdfdisplay` a balíky potrebné na spracovanie jednotlivých častí napíšeme do prostredia `\psinputs`.

Samotná práca pdfTricks vyzerá nasledovne. Každú označenú časť skopíruje do nového `.tex` súboru, ktorý používa balíky uvedené v `\psinputs`. Pomocou "shell escape" postupne zavolá programy `latex`, `dvips`, `epstopdf` a vytvorí z neho DVI, PostScript a nakoniec PDF. Toto PDF následne vloží do dokumentu namiesto `\pdfdisplay` prostredia. Tento postup v zjednodušenej podobe zobrazuje obrázok 4.1.

S takto označeným dokumentom však nastáva problém, ak by sme chceli ako výstup DVI a nie PDF. V takom prípade by sme potrebovali, aby bolo prostredie `\pdfdisplay` ignorované. To sa dá zabezpečiť takto:

- Pomocou balíka `ifpdf` zistíme, či je výstupom PDF alebo DVI.
- Ak je výstupom **PDF** - správame sa ako doteraz.
- Ak je výstupom **DVI** - **ne**načítame pdfTricks, načítame všetky balíky, ktoré sú v `\psinputs` a definujeme prázdne prostredia `\pdfdisplay` a `\pdfpic`.

Nasledujúci kód je prevzatý z <http://tug.org/PSTricks/main.cgi?file=pdf/pdfoutput>

```
\usepackage{ifpdf}
\ifpdf%
  \usepackage{pdftricks}
  \begin{psinputs}
    \usepackage{pstricks}
  \end{psinputs}
\else
  \usepackage{pstricks}
  \newenvironment{pdfdisplay}{}{}
  \newenvironment{pdfpic}{}{}
\fi
```

Pre lepšiu predstavu pripájame aj obrázok 4.2, kde je vidno rozdiely medzi pôvodným dokumentom používajúcim len PSTricks a upraveným dokumentom pre pdfTricks.

4.1.2 Návrhy na vylepšenia

Upravenú verziu súboru pdftricks.sty s navrhovanými zmenami nájde čitateľ na priloženom CD. Presnú lokáciu na CD nájde v tabuľke A.5 v dodatku A. Zmeny boli vykonané a otestované na verzii 1.16, ktorá je súčasťou T_EX Live 2007.

Test funkčnosti \write18

Jednou z prvých vecí, ktoré balík pdfTricks robí, je, že otestuje funkčnosť príkazu \write18 nasledovným spôsobom:

```
\ifPDFTshell
  \def\tmpfile{/tmp/w18-test-\the\year\the\month\the\day\the\time}
  \ifmiktex%
    \immediate\write18{rem >"\tmpfile"}
  \else
    \immediate\write18{touch \tmpfile}
  \fi
\ifmiktex
  \IfFileExists{\tmpfile.}{\PDFTshelltrue}{\PDFTshellfalse}
\else
  \IfFileExists{\tmpfile}{\PDFTshelltrue}{\PDFTshellfalse}
\fi
\fi
```

PSTricks verzia	pdfTricks verzia
<pre> \documentclass{article} \usepackage{slovak} \usepackage{pstricks,pst-text} \begin{document} \begin{pspicture}(-2,-2.5)(2,2.5) \psset{linewidth=0.2pt} \pstextpath(0,1mm){\pscircle{2}}% {\LARGE Ukážka PSTricks a pdfTricks.} \end{pspicture} \end{document} </pre>	<pre> \documentclass{article} \usepackage{slovak,ifpdf} \ifpdf% \usepackage[miktex]{pdftricks} \begin{psinputs} \usepackage{slovak,pstricks,pst-text} \end{psinputs} \else \usepackage{pstricks,pst-text} \newenvironment{pdfdisplay}{}{} \fi \begin{document} \begin{pdfdisplay} \begin{pspicture}(-2,-2.5)(2,2.5) \psset{linewidth=0.2pt} \pstextpath(0,1mm){\pscircle{2}}% {\LARGE Ukážka PSTricks a pdfTricks.} \end{pspicture} \end{pdfdisplay} \end{document} </pre>

Obr. 4.2: Porovnanie dokumentov pre PSTricks a pdfTricks

Myšlienka tohoto testu je, že sa pokúsi pomocou `\write18` vytvoriť na disku súbor s názvom definovaným v `\tmpfile`. V operačnom systéme Microsoft Windows XP však `/tmp/` nie je platná cesta, a preto príkaz `rem >"\tmpfile"` vyhlási chybu a súbor nevytvorí. Následne samozrejme príkaz `\IfFileExists` tento súbor nenájde a mylne usúdi, že `\write18` nefunguje.

V tomto operačnom systéme sa štandardný `Temp` adresár nachádza v `C:/Temp/`. Ďalej vidíme, že sa dvakrát za sebou testuje tá istá podmienka, a teda tieto testy môže zlúčiť. Preto môžeme vyššie uvedený `TEX`ový kód prepísať do nasledovnej podoby:

```

\ifPDFTshell
\def\tmpfile{w18-test-\the\year\the\month\the\day\the\time}
\ifmiktex%
\immediate\write18{rem >"C:/Temp/\tmpfile"}

```

```

\IfFileExists{C:/Temp/\tmpfile.}{\PDFTshelltrue}{\PDFTshellfalse}
\else
\immediate\write18{touch /tmp/\tmpfile}
\IfFileExists{/tmp/\tmpfile}{\PDFTshelltrue}{\PDFTshellfalse}
\fi
\fi

```

Pozornému čitateľovi však iste neuniklo, že v teste používame podmienku `\ifmiktex`. Je nutné si uvedomiť, že príkazy operačných systémov Windows a Linux sú odlišné, a preto potrebujeme celý test rozdeliť na dve časti. MikTeX je známa distribúcia pre Windows, a preto autori balíka použili práve tento názov na rozlíšenie operačných systémov. Pravdivostná hodnota `\ifmiktex` je nastavená pri načítaní balíka, a preto ho musíme v operačnom systéme Windows načítať s nastavením `miktex`.

Slovenčina v grafike

Nevýhodou momentálnej implementácie je, že argumenty príkazov `\write18` sú natvrdo napísané, a tým je aj určené, ktoré programy sa volajú na spracovanie vstupných súborov. Toto môže byť veľký problém, ak grafika obsahuje slovenský alebo český text. Vtedy totiž potrebujeme zavolať `cslatex` s parametrom `-default-translate-file=cp1250cs` pre slovenské kódovanie. Preto je nutné prepísať v súbore `pdftricks.sty` nasledovný riadok:

```

\immediate\write18{latex -interaction=batchmode \PDFTfigname}

na

\immediate\write18{cslatex -interaction=batchmode -translate-file=cp1250cs
\PDFTfigname}

```

Nové nastavenie balíka clean

`pdfTricks` vytvorí ku každému obrázku ďalších 7 súborov, čo môže byť v prípade väčšej práce nepríjemné. Preto by bolo vhodné mať k dispozícii možnosť ich aj jednoducho zmazať. Nemôžeme ich však zmazať vždy, lebo to by znamenalo, že by sme vždy museli znovu spracovávať aj každý obrázok a tým by stratilo zmysel nastavenie `NoProcess`. Preto je vhodné použiť toto nastavenie až pri záverečnom generovaní celého dokumentu. Môžeme to implementovať v niekoľkých krokoch nasledovne:

1. Zavedieme novú podmienku:

```
\newif\ifclean \cleanfalse
```

2. Deklarujeme nové nastavenie:

```
\DeclareOption{clean}{\cleantrue}
```

3. Dopíšeme kód na zmazanie vytvorených súborov, ak používame pdfTricks s týmto nastavením:

```
\ifmiktex%
  \immediate\write18{del \PDFTfigname.aux \PDFTfigname.dvi
    \PDFTfigname.log \PDFTfigname.eps}
  \ifclean%
    \immediate\write18{del \PDFTfigname.pdf \PDFTfigname.ps
      \PDFTfigname.tex}
  \fi
\else
  \immediate\write18{rm \PDFTfigname.aux \PDFTfigname.dvi
    \PDFTfigname.log \PDFTfigname.eps} %%%% LDL-5
  \ifclean%
    \immediate\write18{rm \PDFTfigname.pdf \PDFTfigname.ps
      \PDFTfigname.tex}
  \fi
\fi
```

4.2 ps4pdf

Tento balík pracuje odlišne ako pdfTricks. Na svoju prácu používa balík `preview`, ktorý dokáže vytiahnuť všetky označené časti TeXového dokumentu a uloží ich do DVI súboru, pričom každá takáto časť sa nachádza na samostatnej strane. Takto uložené DVI sa skonvertuje na PDF a každá strana tohoto PDF nahradí jednu označenú časť výsledného dokumentu. To sa dá spraviť veľmi ľahko, keďže obrázky v tomto PDF sú uložené v rovnakom poradí ako označené časti vo vstupnom dokumente. Myšlienka tohoto balíka teda je, že toto PDF bude slúžiť ako akýsi kontajner, kde si uložíme všetky obrázky a neskôr ich odtiaľ budeme postupne vkladať naspäť do dokumentu.

Autorom tohoto balíka je Rolf Niepraschk. V súčasnosti sa však odporúča používať novšiu verziu s názvom `pst-pdf`.

4.2.1 Používanie

Najskôr príkazom

```
\usepackage{ps4pdf}
```

načítame tento balík. Podobne ako pri práci s `pdfTricks`, aj tu musíme všetku PostScriptovú grafiku nejako označiť. Na to slúži makro

```
\PSforPDF.
```

Čitateľovi by malo byť zrejmé, že aj všetky balíky potrebné na vytvorenie PostScriptovej grafiky sa musia nachádzať v tomto makre, inak by nebolo možné túto grafiku vytvoriť a skonvertovať na PDF.

Skutočná práca tohoto balíka prebieha nasledovne:

1. V prvom \LaTeX ovom prechode `preview-latex` vyberie všetky označené časti dokumentu a uloží ich do DVI súboru, pričom každá časť je na samostatnej strane.
2. DVI súbor je skonvertovaný pomocou `dvips`.
3. Tento PostScriptový súbor je následne pomocou `ps2pdf`¹ skonvertovaný na PDF.
4. Nasleduje druhý prechod `pdf\LaTeX`om, kedy je každá označená časť nahradená jednou stranou z tohoto PDF. V tomto momente je tiež možné spustiť iný program ako napríklad `BibTeX`.

Na uľahčenie práce je k tomuto balíku priložený aj skript `ps4pdf`, ktorý vykoná prvé tri kroky z tohoto postupu. Pre lepšie pochopenie sme opäť pridali obrázok 4.3 porovnávajúci dokumenty len pre balík `PSTricks` a `ps4pdf`. Ďalšie užitočné informácie čitateľ nájde v [NV01].

4.3 pst-pdf

Ako sme už spomenuli, tento balík je len novšia a odporúčaná verzia balíka `ps4pdf`. Vznikol v roku 2005 a autormi sú opäť Rolf Niepraschk a Hubert Gäßlein. V princípe tento balík funguje rovnako ako jeho predchodca, takže sa mu príliš venovať nebudeme. Podotkneme len, že už nie je potrebné používať

¹súčasť Ghostscriptu

PSTricks verzia	ps4pdf verzia
<pre> \documentclass{article} \usepackage{slovak} \usepackage{pstricks,pst-text} \begin{document} \begin{pspicture}(-2,-2.5)(2,2.5) \psset{linewidth=0.2pt} \pstextpath(0,1mm){\pscircle{2}}% {\LARGE Ukážka PSTricks a pdfTricks.} \end{pspicture} \end{document} </pre>	<pre> \documentclass{article} \usepackage{slovak} \usepackage{ps4pdf} \PSforPDF{ \usepackage{slovak,pstricks,pst-text} } \begin{document} \PSforPDF{ \begin{pspicture}(-2,-2.5)(2,2.5) \psset{linewidth=0.2pt} \pstextpath(0,1mm){\pscircle{2}}% {\LARGE Ukážka PSTricks a pdfTricks.} \end{pspicture} } \end{document} </pre>

Obr. 4.3: Porovnanie dokumentov pre PSTricks a ps4pdf

makro `\PSforPDF`, ale priamo prostredie `\pspicture` resp. `\postscript`. Balík ponúka aj niekoľko nových nastavení. V prípade ďalšieho záujmu nájde čitateľ podrobnejšie informácie v [NG06].

Kapitola 5

Použitie pdfTricks a ps4pdf v praxi

V tejto kapitole si ukážeme použitie pdfTricks a ps4pdf na niektorých známych balíkoch používajúcich PostScript. Všetky uvedené príklady boli testované v operačnom systéme Microsoft Windows XP s distribúciou T_EX Live 2007. Všetky príklady sa nachádzajú aj na priloženom CD. Kompletný zoznam čitateľ nájde v dodatku A v tabuľke A.3.

5.1 GasT_EX

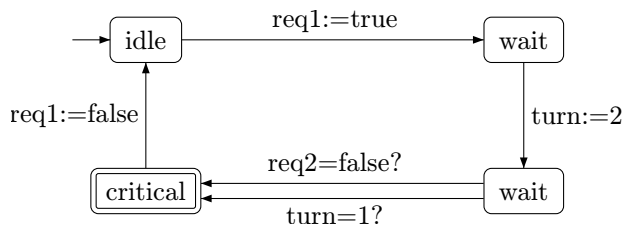
GasT_EX (Graphs and Automata Simplified in T_EX) je balík L^AT_EXových makier na kreslenie grafov, automatov, sietí a diagramov v L^AT_EXovom prostredí `\picture`. Autorom je Paul Gustin. Balík na svoju prácu používa PostScript, a preto ho nie je možné použiť v pdfT_EXu. Čitateľ môže tento balík nájsť na webovej stránke autora: <http://www.lsv.ens-cachan.fr/~gustin/gastex/gastex.html>

Použitie GasT_EXu s pdfTricks je bezproblémové. Pri ps4pdf je to už trochu komplikovanejšie. GasT_EX totiž používa hlavičkový (angl. header) súbor `gastex.pro`, ktorý treba pripojiť pomocou príkazu

```
\pstheader{gastex.pro}.
```

Modifikované dokumenty pre oba balíky sú na obrázku 5.1. Výsledok balíka pdfTricks je na obrázku 5.2. Ďalšie príklady sa nachádzajú na CD.

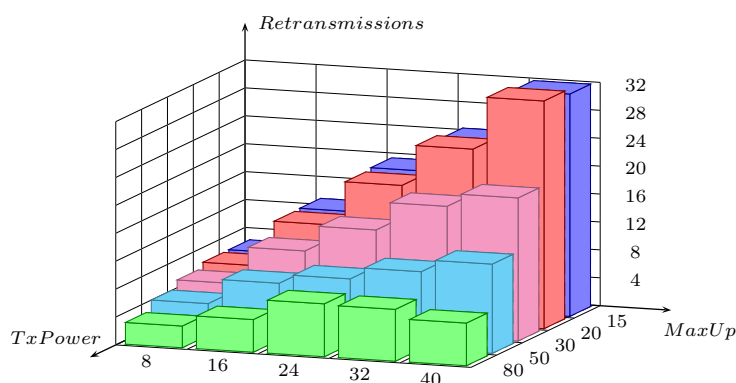
pdfTricks verzia	ps4pdf verzia
<pre> \documentclass{article} \usepackage{slovak,ifpdf} \ifpdf% \usepackage[miktex]{pdftricks} \begin{psinputs} \usepackage[usenames]{color} \usepackage{slovak,gastex} \end{psinputs} \else \usepackage[usenames]{color} \usepackage{gastex} \newenvironment{pdfdisplay}{}{} \fi \begin{document} \begin{pdfdisplay} \begin{picture} \gasset{Nadjust=w,Nadjustdist=2, Nh=6,Nmr=1} \node[Nmarks=i](A)(0,0){idle} \node(B)(50,0){wait} \node(C)(50,-20){wait} \node[Nmarks=r](D)(0,-20){critical} \drawedge(A,B){req1:=true} \drawedge(B,C){turn:=2} \drawedge[syo=-1,eyo=-1](C,D){turn=1?} \drawedge[syo=1,eyo=1,ELside=r](C,D) {req2=false?} \drawedge(D,A){req1:=false} \end{picture} \end{pdfdisplay} \end{document} </pre>	<pre> \documentclass{article} \usepackage{slovak,ps4pdf} \PSforPDF{ \usepackage[noxcolor]{pstricks} \usepackage{slovak,gastex} \pstheader{gastex.pro} } \begin{document} \PSforPDF{ \begin{picture} \gasset{Nadjust=w,Nadjustdist=2, Nh=6,Nmr=1} \node[Nmarks=i](A)(0,0){idle} \node(B)(50,0){wait} \node(C)(50,-20){wait} \node[Nmarks=r](D)(0,-20){critical} \drawedge(A,B){req1:=true} \drawedge(B,C){turn:=2} \drawedge[syo=-1,eyo=-1](C,D){turn=1?} \drawedge[syo=1,eyo=1,ELside=r](C,D) {req2=false?} \drawedge(D,A){req1:=false} \end{picture} } \end{document} </pre>

Obr. 5.1: GasT_EXový dokument pre pdfTricks a ps4pdf

Obr. 5.2: Zodpovedajúci výstup k obrázku 5.1 pre pdfTricks

5.2 pst-3dplot

Tento balík je rozšírením `PSTricks`. Špecializuje sa na kreslenie 3D grafov funkcií. Podrobnosti môže čitateľ nájsť v dokumentácii [Vos07]. Obrázok 5.3 je príkladom tohoto balíka. Kód k tomuto obrázku kvôli jeho dĺžke neuvádzame, čitateľ ho nájde na priloženom CD. Presná lokácia na CD je v tabuľke A.3.



Obr. 5.3: Ukážka `pst-3dplot` s `pdfTricks`

5.3 Beamer

`Beamer`[Tan07] je balík určený na tvorbu prezentácií a slajdov. Autorom je Till Tantau. Použitie vo forme prezentácie nájde čitateľ na priloženom CD. Testovali sme `GasTeX` s `pdfTricks`, `ps4pdf` aj `pst-pdf`. V jednotlivých prezentáciach sú aj v krátkosti popísané potrebné kroky na použitie s týmto balíkom.

Kapitola 6

Záver

Cieľom predkladanej bakalárskej práce bolo preskúmať a zrealizovať možnosti používania PostScriptových makier v programe pdf \TeX . Najskôr sme popísali formáty PostScript a PDF a vysvetlili sme, aké výhody prináša ich vzájomná kombinácia. Vysvetlili sme akým spôsobom sa v \TeX u pracuje s PostScriptovým kódom a stručne sme sa oboznámili s najpoužívanejším balíkom PostScriptových makier a jeho niekoľkými rozšíreniami, ktoré prácu s PostScriptom výrazne uľahčujú. Nakoniec sme prebrali dve najpoužívanejšie dostupné riešenia, balíky pdfTricks a ps4pdf resp. novší pst-pdf. Stručne sme ukázali ako ich používať, pričom sme navrhli a aj implementovali niekoľko vylepšení.

Podrobne sme analyzovali danú problematiku, takže po prečítaní by čitateľ mal mať aspoň základnú predstavu o tom ako pracujú programy \TeX a pdf \TeX , ako z \TeX ového dokumentu vzniká čitateľný PostScriptový dokument a ako oddeliť PostScriptové časti od zvyšku dokumentu. Ďalej by mu malo byť zrejmé, aký je rozdiel medzi PDF vytvoreným priamo pdf \TeX om a PDF skonverovaným z PostScriptu.

V celej práci sme predpokladali len základné vedomosti z oblasti \TeX u, a tak ani menej skúsenému \TeX ovému užívateľovi by nemalo robiť ťažkosti túto problematiku pochopiť a spomínané balíky si rýchlo a ľahko osvojiť a úspešne používať.

Práca má slúžiť ako krátky ale rýchly prehľad danej problematiky, a preto nevysvetľuje všetko do úplných detailov. Namiesto toho sú v každej kapitole a časti uvedené odkazy na ďalšiu, oveľa podrobnejšiu literatúru, v ktorej v prípade hlbšieho záujmu môže čitateľ nájsť dodatočné informácie. Ukázali sme si aj použitie v praxi, takže práca môže čitateľovi slúžiť aj ako príručka.

Cieľom práce je tiež rozšíriť prehľad a predstavu čitateľa o tom, čo všetko $\text{T}_{\text{E}}\text{X}$ pomocou niekoľkých jednoduchých príkazov dokáže. Veľa ľudí zvyknutých na WYSIWYG (What You See Is What You Get, čo vidíš to dostaneš) editory ako sú napríklad Microsoft Word môže byť relatívnou komplikovanosťou $\text{T}_{\text{E}}\text{X}$ u ľahko odradených. Práca by preto mala tiež motivovať čitateľa, aby využíval výhody oboch formátov zároveň a vytváral tak vysoko kvalitné elektronické dokumenty alebo ako povedal Knuth, krásne knihy—a hlavne knihy obsahujúce množstvo matematiky.

Dodatok A

Obsah priloženého CD

Na priloženom CD je uložená táto bakalárska práca v elektronickej podobe. Okrem toho je na ňom uložená všetka citovaná literatúra, ktorá bola dostupná v elektronickej podobe ako aj príklady odkazované v práci.

Tabuľka A.1: Adresár /bin.

Názov súboru	Popis
Gatial.pdf	Táto práca vo formáte PDF.

Tabuľka A.2: Adresár /doc – citovaná literatúra.

Názov súboru	Odkaz
beameruserguide.pdf	[Tan07]
euclide_english.pdf	[Rod05]
gentle.pdf	[Doo87]
manual-1.16.pdf	[RR03]
pst-3dplot-doc.pdf	[Vos07]
pst-pdf.pdf	[NG06]
pstricks-doc.pdf	[Zan03]

Tabuľka A.3: Adresár /example – príklady odkazované v práci.

Názov súboru	Popis
beamer.zip	Obsahuje príklady používania balíka Beamer s GasT _E Xom. Skomprimované programom WinZip.
beamer.rar	Obsahuje príklady používania balíka Beamer s GasT _E Xom. Skomprimované programom WinRar.
gastex.zip	Obsahuje príklady používania balíka GasT _E X s pdfTricks, ps4pdf a pst-pdf. Skomprimované programom WinZip.
gastex.rar	Obsahuje príklady používania balíka GasT _E X s pdfTricks, ps4pdf a pst-pdf. Skomprimované programom WinRar.
pst-3dplot-example.tex	Zdrojový kód k obrázku 5.3.
pst-3dplot-example.pdf	Obrázok 5.3 vo formáte PDF.

Tabuľka A.4: Adresár /source – zdrojový kód práce.

Názov súboru	Popis
source.zip	Obsahuje zdrojový kód celej práce. Skomprimované programom WinZip.
source.rar	Obsahuje zdrojový kód celej práce. Skomprimované programom WinRar.

Tabuľka A.5: Adresár /zmeny – súbory s navrhovanými zmenami.

Názov súboru	Popis
pdftricks.sty	Upravená verzia zdrojového kódu pre balík pdfTricks s navrhovanými zmenami.

Literatúra

- [Doo87] Michael Doob. *A Gentle Introduction to T_EX: A Manual for Self-Study*. 1987. <http://ctan.tug.org/tex-archive/info/gentle>.
- [Inc85] Adobe Systems Incorporated. *PostScript language tutorial and cookbook*. Addison-Wesley Publishing Company, 2nd, edition, 1985.
- [Inc99] Adobe Systems Incorporated. *PostScript language reference manual*. Addison-Wesley Publishing Company, 3rd, edition, 1999. <http://www.adobe.com/products/postscript/pdfs/PLRM.pdf>.
- [Inc06] Adobe Systems Incorporated. *PDF Reference, Adobe Portable Document Format version 1.7*. 6th, edition, 2006. http://www.adobe.com/devnet/acrobat/pdfs/pdf_reference.pdf.
- [Knu84] Donal Ervin Knuth. *The T_EXbook*. Addison-Wesley Publishing Company, 20th, edition, 1984. <http://www.ctan.org/tex-archive/help/Catalogue/entries/texbook.html>.
- [NG06] Rolf Niepraschk and Hubert Gäßlein. *The pst-pdf package*. 2006. <http://www.ctan.org/tex-archive/help/Catalogue/entries/pst-pdf.html>.
- [NV01] Rolf Niepraschk and Herbert Voß. *The package ps4pdf: from PostScript to PDF*. *TUGboat*, 22(4):290–292, 2001. <http://www.ctan.org/tex-archive/help/Catalogue/entries/ps4pdf.html>.
- [Rei90] Glenn C. Reid. *Thinking in PostScript*. Addison-Wesley Publishing Company, 1990. <http://www.rightbrain.com/download/books/ThinkingInPostScript.pdf>.

- [Rod05] Dominique Rodriguez. *The pst-euclide Package, version 1.3.2*. 2005. <http://www.ctan.org/tex-archive/help/Catalogue/entries/pst-eucl.html>.
- [RR03] C. V. Radhakrishnan and C. V. Rajagopal. *Trivial Experiments with psTricks manipulation, version 1.16*. 2003. <http://www.ctan.org/tex-archive/help/Catalogue/entries/pdftricks.html>.
- [Tan07] Till Tantau. *The beamer class, Manual for version 3.07*. 2007. <http://latex-beamer.sourceforge.net/>.
- [Vos07] Herbert Voss. *3D plots: PST-3dplot v1.72, Documentation*. 2007.
- [Zan03] Timothy Van Zandt. *PSTricks, PostScript macros for Generic TeX, Users Guide, version 97*. 2003. <http://tug.org/PSTricks/main.cgi/>.