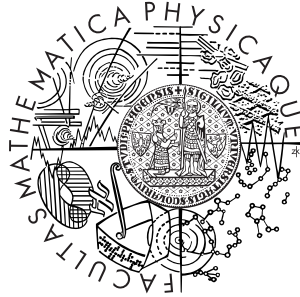


Univerzita Karlova v Praze  
Matematicko-fyzikální fakulta

# BAKALÁŘSKÁ PRÁCE



Aleš Šnupárek

## Předtisková příprava dokumentů

Katedra aplikované matematiky

Vedoucí bakalářské práce: Mgr. Martin Mareš  
Studijní program: Informatika, programování

2007

Na tomto místě bych chtěl poděkovat lidem, bez jejichž pomoci by nikdy tato práce nevznikla. Mé díky patří zejména Mgr. Martinu Marešovi za vedení bakalářské práce, za dobré rady a věcné připomínky.

Prohlašuji, že jsem svou bakalářskou práci napsal samostatně a výhradně s použitím citovaných pramenů. Souhlasím se zapůjčováním práce a jejím zveřejňováním.

V Praze dne 14. prosince 2007

Aleš Šnupárek

# Obsah

<b>1</b>	<b>Úvod</b>	<b>5</b>
<b>2</b>	<b>Dokumenty a práce s nimi</b>	<b>6</b>
2.1	Dokument . . . . .	6
2.2	Stránka . . . . .	6
2.3	Formáty dokumentů . . . . .	6
2.4	Předtiskové úpravy . . . . .	7
<b>3</b>	<b>Vybrané formáty</b>	<b>8</b>
3.1	Grafický meta-formát PostScript . . . . .	8
3.2	Afinní transformace . . . . .	9
3.3	DSC komentáře . . . . .	9
3.4	Portable Document Format . . . . .	10
<b>4</b>	<b>Současné programy</b>	<b>12</b>
4.1	PSUtils . . . . .	12
4.2	Pdftk . . . . .	12
<b>5</b>	<b>Pspdftool</b>	<b>13</b>
5.1	Uživatelská příručka . . . . .	13
5.1.1	Popis „užitečných příkazů“ . . . . .	15
<b>6</b>	<b>Pspdftool - implemntace</b>	<b>20</b>
6.1	Zobecnění dokumentu . . . . .	20
6.2	Část nezávislá na konkrétním formátu . . . . .	20
6.3	Implementace konkrétních formátů . . . . .	21
6.3.1	PostScript . . . . .	21
6.3.2	PDF . . . . .	21
6.4	Interpret příkazů . . . . .	22
<b>7</b>	<b>Závěr</b>	<b>23</b>
	<b>Literatura</b>	<b>24</b>
<b>A</b>	<b>Obsah přiloženého CD-ROM</b>	<b>25</b>

Název práce: Předtisková příprava dokumentů  
Autor: Aleš Šnupárek  
Katedra (ústav): Katedra aplikované matematiky  
Vedoucí bakalářské práce: Mgr. Martin Mareš  
e-mail vedoucího: mares@kam.mff.cuni.cz

Abstrakt: V předložené práci se zabývám některými tiskovými formáty (PDF, PostScript) a možnostmi úprav hotových dokumentů před samotným tiskem. Dále zkoumám současné nástroje pro dané úpravy. V části se zabývám návrhem a realizací nástroje, který umožňuje provádět některé ze základních předtiskových úprav.

Klíčová slova: PostScript, PDF, tisk

Title: Preprinting preparation of documents  
Author: Aleš Šnupárek  
Department: Department of Applied Mathematics  
Supervisor: Mgr. Martin Mareš  
Supervisor's e-mail address: mares@kam.mff.cuni.cz

Abstract: In the present work I study printing formats (PDF, PostScript) and abilities of modifying already finished documents before printing. In the other part of my work I focus on commonly used tools for pre-print preparations of documents. Last part of my work is design and implementation of tool for pre-print preparations of documents.

Keywords: PostScript, PDF, printing

# Kapitola 1

## Úvod

V této práci „Předtisková příprava dokumentů“ zkoumám možnosti úprav dokumentů v tiskovém formátu. Nezabývám se nástroji pro tvorbu těchto dokumentů, ani nezkoumám různé tiskové metody. Zabývám se pouze dokumenty v tiskovém formátu a to konkrétně v PostScriptu a v PDF.

Práce je rozdělena do několika částí. První část vysvětluje základní pojmy, které se týkají dokumentů a předtiskových úprav. Druhá část popisuje jak fungují formáty PostScript a PDF s ohledem na jejich předtiskové úpravy. V třetí části je popis některých současných nástrojů pro předtiskové úpravy dokumentů. Poslední část se zabývá mnou vytvořeným nástrojem pspdftool, který umí provádět různé předtiskové úpravy ve formátech PostScript a PDF. Příkladem těchto úprav může být: změna rozměrů stránky, selekce a přeuspořádání pořadí stránek nebo umístění více stránek na jeden arch papíru.

# Kapitola 2

## Dokumenty a práce s nimi

### 2.1 Dokument

Pod pojmem dokument rozumíme soubor na paměťovém médiu uchováající nějakou informaci. Na dokument určený primárně pro tisk klademe následující požadavky: musí definovat jednotlivé stránky, definovat jejich obsah a uchovávat jejich atributy.

### 2.2 Stránka

Stránka je soubor objektů, které se vytisknou na jeden arch papíru. Objektem může být text, čára, křivka, bitmapa, nebo jiný grafický útvar.

Pro umístění objektu na stránce je potřeba definovat souřadnicový systém. Zpravidla se používá kartézský souřadnicový systém, který má počátek v levém dolním rohu stránky. Jeho x-ová souřadnice je orientovaná vodorovně, zleva doprava, y-ová souřadnice je orientovaná svisle, směrem zdola nahoru. Základní jednotkou tohoto systému může být tiskařský bod.<sup>1</sup>

Mezi atributy popisující konkrétní stránku patří: velikost papíru, orientace textu a bounding box (bbox). Bounding box určuje nejmenší obdélníkovou oblast, do které se vykreslí všechny objekty z příslušné stránky.

### 2.3 Formáty dokumentů

Formát dokumentu určuje, jaké informace a jakým způsobem budou reprezentovány v souboru.

Pro vytvoření a úpravu dokumentů se ve většině případů používají patřičné editory (textové, grafické a další). Pokud lze dokument vytisknout přímo z editoru, obsahuje příslušný editor funkci, která vygeneruje z upravovaného souboru výstup do vhodného formátu pro tisk. Dle tohoto hlediska můžeme rozdělit formáty na následující dva druhy: formáty editorů a formáty určené k tisku.

Formát z editoru nemusí obsahovat dostatečné informace pro výsledný tisk. U bitmapových formátů často chybí údaj o rozměru papíru, o jeho orientaci, kvalitě tisku a podobné.

---

<sup>1</sup>1 bod obvykle odpovídá 1/72 palce

Dalším nedostatkem formátu editoru je případná závislost na konkrétním tiskovém prostředí (rozlišení tiskárny, dostupná písma, a podobně). Například pro korektní zobrazení dokumentu z OpenOffice je potřeba mít nainstalované řezy písma, která jsou vyžadována daným dokumentem.

Formáty pro tisk, na rozdíl od formátů editorových, nejsou určeny k další editaci. Tisknutelným formátem může být jazyk určený přímo pro tiskárnu (PostScript, PCL, GDI, HPGL). Převod dokumentu na tento jazyk může mít jistá omezení, například závislost na možnostech konkrétním typu (rozlišení, maximální rozměr papíru a další). Tisknutelným formátem může být také obecný formát obsahující informace určené pro tisk (například PDF). Ač formáty pro tisk nelze obecně editovat, lze na nich provádět některé operace, které budou popsány dále. V následujícím textu se budu zabývat převážně formáty PostScript a PDF.

## 2.4 Předtiskové úpravy

Předtiskové úpravy se týkají formátů explicitně určených pro tisk. Patří mezi ně přeskupení, otočení, selekce, změna formátu papíru, zvětšení či zmenšení stránek, ale netýkají se: změny obsahu, stylu odstavce nebo jiný zásah do struktury stránky.

Důvodem pro předtiskové úpravy může být absence dané úpravy v aplikaci, ve které dokument vznikl, nebo nedostupnost podkladů, ze kterých vznikl dokument v tisknutelném formátu. Příkladem mohou být dokumenty od třetí strany, jako jsou různé dokumentace.

Některé ovladače tiskáren umožňují provést některé z těchto úprav přímo, ale vždy se jedná o proprietární rozšíření konkrétního ovladače. Proto je vhodné mít možnost tyto úpravy nezávisle provést pomocí nějakého nástroje.

# Kapitola 3

## Vybrané formáty

V této kapitole budou popsány formáty PostScript a PDF, které jsou vhodné pro tisk a je na nich možné realizovat některé z předtiskových úprav.

### 3.1 Grafický meta-formát PostScript

PostScript[1] je programovací jazyk navržený firmou Adobe. Je primárně určen pro tisk dokumentů na laserových tiskárnách, ale může být použit pro popis grafiky i na jiném zařízení (například obrazovce počítače). Pro jeho převod do rastrové grafiky se může použít GhostScript[4].

PostScript je turingovsky úplný programovací jazyk s prefixovou notací. Je podobný svou syntaxí jazyku Forth, ale na rozdíl od něj je specializovaný pro popis grafického výstupu. Mezi základní elementy jazyka PostScript patří: oblouk, Bèzierova křivka a úsečka. Pro písmo existují speciální příkazy, které vykreslí písmena z křivek. Tyto elementy nejsou závislé na rozlišení výstupního zařízení.

PostScriptový interpret využívá několik zásobníků. Mezi nejdůležitější patří zásobník na operandy příkazů a zásobník pro uložení slovníků. Slovník je kolekce párů klíč a hodnota. Přidáním nového záznamu do slovníku lze definovat pojmenované konstanty nebo nové příkazy, případně předefinovat stávající příkazy.

PostScript pracuje s dvěma souřadnicovými systémy: device space a user space. Device space je souřadnicový systém cílového výstupního zařízení. User space je souřadnicový systém, ve kterém se zadává poloha objektů (například čar). Počátek tohoto systému je v levém dolním rohu. X-ová osa je orientována zleva do prava a y-ová osa vede zdola nahoru. Vzdálenost dvou bodů (tiskařských bodů) v tomto systému je 1/72 palce. Převod mezi user space a device space je realizován pomocí transformační matice, která reprezentuje afinní transformaci.

Cesta je kolekce čar a křivek umístěných na stránce, ale neobsahuje informaci o tom, jakým způsobem bude vykreslena. Aktuální cesta je právě rozkreslená cesta. Cesta může definovat ořez, tj. oblast, za kterou se nebude vykreslovat, nebo být křivkou určité šířky a barvy.

Graphics State jsou různá nastavení popisující stav grafického systému. Mezi ně patří: aktuální cesta, aktuální ořezová cesta, aktuální font, aktuální transformační matice, barva,



typ vykreslované čáry. Tento stav je možné uložit, respektive obnovit, pomocí operátorů *gsave* a *grestore*.

## 3.2 Afinní transformace

Afinní transformace se aplikuje na jednotlivé grafické objekty. Touto transformací lze popsat všechny běžné transformace, jako je posunutí, otočení nebo změna velikosti. Lze jí reprezentovat pomocí transformační maticí  $3 \times 3$ . Tuto reprezentaci používá PostScript a PDF. Pro bod se souřadnicemi  $[x, y]$  vypadá následovně.

$$[x', y', 1] = [x, y, 1] \times \begin{pmatrix} a & b & 0 \\ c & d & 0 \\ e & f & 1 \end{pmatrix}$$

Jednotlivé transformace jsou uzavřené na skládání a v případě maticové reprezentace se jedná o násobení matic.

## 3.3 DSC komentáře

PostScript nedefinuje žádnou strukturu dokumentu, nelze tedy bez jeho interpretace rozpoznat jednotlivé stránky. Tento problém vyřešila firma Adobe zavedením speciálních komentářů DSC (document structure comment)[3]. Komentář v PS souboru začíná znakem `%` a končí koncem řádky. DSC komentář začíná na začátku řádky znaky `%%`. Díky těmto komentářům lze rozdělit dokument na následující části: prolog a script.

Do prologu patří header section, defaults section a procedure section. Část Script můžeme rozdělit na část document setup, samostatné stránky a trailer.

Na začátku souboru s DSC komentáři je řádek určující verzi DSC, současná verze je 3.0. Po této řádce následuje výše zmíněná hlavička, jedná se o souvislou oblast komentářů (neobsahuje příkazy pro PS interpret). Hlavička končí komentářem `%%EndComments`, nebo prázdným řádkem. V hlavičce jsou uvedeny základní informace o dokumentu, jako jsou: počet stránek `%%Pages: počet stránek | (atend)`, orientace textu na stránce `%%Orientation: Portrait | Landscape`, bounding box, pořadí stránek v dokumentu `%%PageOrder: Ascend | Descend | Special | (atend)`. Symbol (atend) místo konkrétní hodnoty bude nahrazen hodnotou příslušné vlastnosti z traileru na konci souboru.

Defaults section je nepovinná část PS dokumentu, která je složena pouze z DSC komentářů. Obsahuje doplňující informace o dokumentu, určené pro správce tisku.

Procedure section (procdef) může obsahovat definice funkcí, které bude dále program používat. Toto místo je vhodné pro přidání vlastních definic (například nových operátorů).

V sekci document setup může být postscriptový kód pro nastavení tiskárny (velikost papíru, nahrání fontů). Po ní následují jednotlivé stránky.

Začátek stránky označuje komentář `%%Page: jméno_stránky její_číslo`. Při tisku by neměly být stránky závislé na pořadí, v jakém jsou uvedeny v souboru. Samotnou stránku lze obdobně jako celý dokument rozdělit na několik částí: Page Setup, obsah stránky, PageTrailer. Toto dělení je nepovinné a většina programů generujících PostScript ho nepoužívá. U každé

stránky lze za pomoci DSC explicitně vyjádřit rozměr papíru, orientaci textu a bounding box.

Po všech stránkách následuje Trailer. Může obsahovat kód pro uvedení PS interpretu do výchozího stavu před tiskem a může obsahovat podobné informace, jako obsahuje hlavička. Soubor je zakončen řádkem %%EOF.

### Příklad souboru s DSC komentářem:

```
%!PS-Adobe-3.0
%%Creator: dvips(k) 5.78 Copyright 1998 Radical Eye Software ...
%%Title: popisky.dvi
%%Pages: 1
%%PageOrder: Ascend
%%BoundingBox: 0 0 936 86
%%DocumentFonts: CenturyExpdCEBTBoldItalic CenturyExpdCEBTItalic
%%EndComments
%DVIPSCommandLine: dvips -Plino1200 -D1800 -tfilm1 popisky
%DVIPSPParameters: dpi=1800, compressed
%DVIPSSource: TeX output 2001.09.24:2244
... Zde se nacházejí definiční hlavičky ...
%%EndProlog
%%BeginSetup
... Zde se nacházejí nastavovací hlavičky ...
%%EndSetup
%%Page: 1 1
... Zde se nachází popis stránky 1 ...
%%Trailer
... Zde se nachází ukončovací sekvence ...
%%EOF
```

Jazyk PostScript a jeho DSC komentáře jsou obsáhlejší, než je uvedeno v této části, podrobnější informace je možné nalézt ve specifikaci PostScriptu a DSC komentářů.

## 3.4 Portable Document Format

Portable Document Format (PDF)[2] byl vyvinut firmou Adobe. Jeho hlavní výhodou je zachování věrnosti dokumentu při přenosu souboru z jednoho počítače na druhý. Vzhledem k tomu, že některé části PDF mohou být komprimovány, je také vhodný pro přenos dokumentů přes síť Internet. Formát umožňuje nastavit některé bezpečnostní atributy dokumentu, jako jsou: povolení tisku, povolení kopírování z dokumentu, povolení změn v dokumentu a digitální podpis.

PDF definuje následující základní objekty: boolean, celočíselný a reálný typ, řetězec, jméno, pole, slovník, stream a nulový objekt. Objekt typu Boolean může nabývat hodnot true a false. Číselný typ je posloupnost čísel. Reálný typ jsou dvě čísla oddělená desetinnou tečkou. Řetězce mohou být lexikální nebo hexadecimální. Jméno je posloupnost znaků začínající

lomítkem. Pole na rozdíl od jazyku C může obsahovat prvky různého typu. Slovník je seznam dvojic. První prvek z dvojice je jméno a druhý je hodnota. Null objekt reprezentuje prázdný objekt.

Stream je objekt pro uschování libovolné posloupnosti bajtů. Je složen ze slovníku, po kterém následuje samotný bitový stream, který je uvozen identifikátory *stream* a *endstream*. Slovník streamu musí obsahovat povinný atribut *Length*, který udává délku bitového streamu. Nepovinný atribut *Filter* udává způsob, jakým je zakódovaný bitový stream. Uvnitř streamu nemůže být odkaz na ostatní objekty v dokumentu.

V PDF je definován nepřímý objekt (indirect object). Nepřímým objektem může být libovolný objekt. Pro jeho jednoznačnou identifikaci jsou použita dvě čísla: první určuje objekt, druhé určuje jeho revizi, novější verze objektu má vyšší číslo. Syntakticky nepřímý objekt vypadá následovně: *číslo revize popis\_samotného\_objektu endobj*. Na tento objekt se lze odkázat pomocí následující syntaxe, *číslo\_objektu revize R*.

Celý soubor je složen z nepřímých objektů. Pro rychlejší přístup k těmto objektům je v souboru uložena tabulka (cross reference table). Obsahuje pro každý nepřímý objekt v souboru následující záznam: číslo jeho revize a offset od začátku souboru, kde se objekt nachází. Díky této tabulce není potřeba pro přístup ke konkrétnímu objektu celý soubor sekvenčně přečíst. Pozice v souboru, na které se tabulka nachází, je uvedena na následující řádce za identifikátorem startxref. Tento identifikátor následuje za trailerem.

Na konci souboru je sekce trailer. Jedná se o slovník, ve kterém je odkaz na catalog objekt. Pokud byl dokument inkrementálně aktualizován, nachází se v traileru odkaz na předchozí trailer. V catalogu je uveden odkaz na strom stránek, případně informace o zabezpečení souboru a jiné globálně platné informace pro dokument.

Strom stránek je tvořen ze dvou typů objektu: uzlů (Pages, odkazy na další stránky, mohou obsahovat některé společné informace pro stránky v podstromě) a listů (samotné stránky). Objektem stránky je slovník. Obsahuje odkaz na *Content Stream*, popisuje vzhled stránky, resources, přes které se *Content Stream* stránky odkazuje na objekty v dokumentu (např. typ fontu), rozměry papíru (MediaBox) a rozměr bounding boxu (TrimBox).

Mezi často se vyskytující objekt v PDF patří XObject. Jedná se o stream, struktura jeho slovníku je velmi podobná struktuře slovníku popisujícího stránku. Používá se pro popis nějakého grafického elementu. Příkladem jeho použití může být vložení loga na stránku. Toto logo je reprezentováno XObjectem.

Stream pro popis obsahu stránky nebo XObjectu je syntakticky podobný jazyku PostScript, tj. prefixovou notací, pouze operátory mají jiné názvy a nelze je předefinovat. Z grafických operátorů existují křivky, čáry, n-úhelníky, ořezové cesty a transformační matice.

Inkrementální update je způsob aktualizace, kdy se na konec dokumentu přidávají nově aktualizované objekty, nová tabulka referencí a trailer. Při této aktualizaci není potřeba přepisovat celý dokument.

Pro podrobnější specifikaci PDF lze doporučit Adobe PDF reference Manual.

# Kapitola 4

## Současné programy

Tato práce se zabývá převážně programy PSUtils a pdftk. Byly zvoleny, protože jsou dostupné a pracují s formáty popisovanými v této práci. Dále existují m.j. FinePrint, Acrobat. Ty vzhledem k jejich těžší dostupnosti (licence) dále nejsou zmiňovány.

### 4.1 PSUtils

PSUtils[5] jsou sadou nástrojů pro práci s PostScriptovými dokumenty. Mezi tyto programy patří: psnup (uspořádání více stránek na jednu), pselect (výběr jednotlivých stránek z dokumentu), pstops (nástroj pro libovolné transformace), psbook (tvorba tiskových signatur pro tisk bookletů) a psresize (změna rozměrů stránek). PSUtils jsou napsány v jazyce C a obsahují několik skriptů napsaných v Perlu.

PSUtils umí většinu z výše zmíněných předtiskových operací. Nástroj pstops umí kombinovat následující základní operace: změnu velikosti stránky, rotaci stránky o násobky 90 stupňů, posunutí stránky a spojení dvou stránek. Umí spojit několik dokumentů do jednoho, ale pouze takové, které mají stejný prolog. Nelze však s ním provést následující operace: zrcadlení stránek, kreslení či psaní textu na stránku.

### 4.2 Pdftk

Jedná se o nástroj[6] napsaný v jazyce Java. Tímto programem je možné provádět následující operace: spojení více dokumentů do jednoho, výběr stránek z dokumentu, otočení stránek, dekryptovat PDF, enkryptovat PDF, předvyplnění formulářů, opravit poškozené PDF a další specifické úkony pro PDF formát. PDFTk umožňuje provádět úkony, které nepatří mezi předtiskové úpravy. Současným nedostatkem tohoto programu jsou: neschopnost uspořádání více stránek na jeden arch papíru a neschopnost doplnění textu na stránku.

# Kapitola 5

## Pspdftool

Výše zmíněné nástroje pro předtiskovou úpravu dokumentů se vzájemně liší ve svém uživatelském rozhraní. Proto byl vytvořen univerzální nástroj, který bude schopen pracovat se stejným rozhraním a různými formáty. Bude také schopen provádět operace, které současná řešení neobsahují. Nástroj byl navržen s ohledem na možnost jeho rozšíření o další formáty.

### 5.1 Uživatelská příručka

Program `pspdftool` je konzolová aplikace, která se spouští z příkazové řádky. Programu je potřeba předat jako argument příkazy, nebo soubor s příkazy, které bude program provádět na vstupní soubory. Jejich jména následují za příkazy. Posledním argumentem je jméno výstupního souboru. Pokud je uvedeno více vstupních souborů, nejprve jsou spojeny do jednoho, jestliže to daný formát umožňuje. Pokud jsou argumenty reprezentující vstupní a výstupní soubory vynechány, program načte soubor ze standardního vstupu a výsledný soubor vypíše na standardní výstup. Současně podporované formáty jsou PostScript a PDF.

Ze vstupního souboru je vytvořen seznam stránek. Vstupem každého příkazu je seznam stránek, na který se aplikuje příslušná operace. Jeho výstupem je upravený seznam stránek. Příkazy se vykonávají zleva doprava. První příkaz zpracuje seznam ze vstupních souborů a předá ho následujícímu příkazu. Z výsledného seznamu posledního příkazu je vytvořen výstupní soubor. Jednotlivé příkazy se mezi sebou oddělují bílými znaky.<sup>1</sup>

`vstupní_seznam_stránek` → `příkaz` → `výstupní_seznam_stránek`

#### Příklady:

<code>pspdftool "nup(2, paper=a4)" in.pdf out.pdf</code>	načte soubor <code>in.pdf</code> aplikuje na seznam stránek příkaz <code>nup</code> a výsledek uloží do souboru <code>out.pdf</code> (uvozovky kolem příkazu jsou kvůli shellu)
<code>pspdftool -f cmds.txt in.pdf out.pdf</code>	na rozdíl od předchozího se budou číst příkazy ze souboru <code>cmds.txt</code>

---

<sup>1</sup>bílým znakem chápeme: mezeru, tabulátor, novou řádku

## Úžitečné příkazy:

<code>select{1..5}</code>	vybere se prvních pět stránek ze seznamu
<code>scalet(a4)</code>	přizpůsobí velikosti stránky na papír velikosti a4
<code>nup(2)</code>	vyrovnání dvou stránek na jednu
<code>book</code>	přerovnění stránek pro tisk bookletu
<code>apply{-1 del}</code>	smazání pospední stránky v seznamu

Příkaz se skládá ze tří částí: názvu, jeho parametrů a selektoru stránek. Název je posloupnost znaků identifikující typ příkazu.

Sekce parametru příkazu začíná levou a končí pravou kulatou závorkou. Parametry mohou být následujících typů: celočíselného, desetinného, rozměru, řetězce, nebo identifikátoru. Celočíselný typ je posloupnost čísel. Desetinný typ obsahuje celočíselnou a desetinnou část. Obě části jsou reprezentované celočíselným typem a jsou od sebe odděleny desetinnou tečkou. Rozměr obsahuje desetinný typ a identifikátor jednotky. Jednotkou může být: mm, cm, pt a in. Implicitně se pracuje s tiskařským bodem. Řetězec je posloupnost znaků mezi uvozovkami. Identifikátor je posloupnost písmen.

Jednotlivé parametry příkazu jsou mezi sebou odděleny čárkou. Mezi čárkou a parametrem příkazu může být posloupnost bílých znaků. Parametry příkazu je možné zadávat pozičně, nebo je explicitně pojmenovat. Poziční parametry musí vždy být před pojmenovanými.

Některé parametry mohou mít uvedenou výchozí hodnotu. Potom není nutné tento argument explicitně uvádět.

## Příklady:

<code>foo</code>	příkaz bez parametru
<code>foo(a=1)</code>	příkaz <code>foo</code> , kterému je nastaven parametr <code>a</code> na 1
<code>foo(1)</code>	příkaz <code>foo</code> , kterému je první parametr nastaven na 1
<code>foo(a=1, 1)</code>	pojmenovaný argument před pozičním, chybně

Poslední částí je selektor stránek, který začíná levou a končí pravou složenou závorkou. Selektor je skupina výběrů stránek, které jsou mezi sebou odděleny mezerami. Výběr stránek může být číslo, případně interval. Pokud výběr začíná '-', jedná se o výběr z konce seznamu, běžně se indexuje od začátku. Za výběrem může následovat seznam příkazů, které se na něm vykonají.

## Příklady:

<code>1</code>	výběr stánky 1
<code>1..2</code>	výběr stránek 1 až 2 (včetně 2)
<code>-1</code>	výběr poslední stránky ze seznamu
<code>-1..2</code>	výběr poslední a předposlední stránky, „od konce seznamu výběr první a druhé“
<code>foo(a){1..2 bar}</code>	příkaz <code>foo</code> s parametrem <code>a</code> a selektorem <code>1..2</code> , na který se aplikuje příkaz <code>bar</code>
<code>foo{1..2 -1..2}</code>	příkaz <code>foo</code> bez parametru a selektory <code>1..2</code> a <code>-1..2</code>

## Syntaxe příkazu:

`jmeno(pevný_parametr_1, ..., název_parametru=hodnota, ...) {výběr příkazy ...}`

Program používá kartézský souřadnicový systém s počátkem v levém dolním rohu. Osa x je vodorovná orientovaná směrem zleva doprava a osa y má orientaci svislou ze zdola na horu. Rozměry papíru je možné zadat názvem, v tom případě se jedná o identifikátor, nebo explicitně pomocí dvou čísel, šířky a výšky. Název příkazů, u kterých se explicitně zadává rozměr papíru, končí číslovkou 2. Například k příkazu *crop* existuje varianta *crop2*.

### 5.1.1 Popis „užitečných příkazů“

Asi mezi nejčastější předtiskovou úpravou dokumentu patří „transformace n stránek na jeden arch papíru“. K tomu se hodí příkaz *nup*.

<code>nup(2)</code>	otočí stránku o 90 stupňů a přerovná dvě stránky na jednu
<code>nup(4)</code>	přeskládá čtyři stránky na jednu bez rotace
<code>nup(1,2,rotate=90)</code>	je ekvivalent příkazu <code>nup(2)</code> , v x-ové ose 1 stránka (vodorovně) a v y-ové ose 2 stránky (svisle)
<code>nup(2,2,paper=a4)</code>	je ekvivalent příkazu <code>nup(4)</code> , akorát výsledná stránka má formát a4

Další užitečné parametry příkazu *nup* jsou: *frame* (orámování), kde číslo udává šířku čáry, *center* (výchozí on výsledek je centrován, *off* nic se necentruje, *final* centruje se až výsledná stránka), *dx*, *dy* (minimalní šířka mezi stránkami), *orient* (postup kladení stránek na stránku, *portrait* zleva doprava, *z* hora dolů, *landscape* zprava doleva z hora dolů). Pokud není *orient* nastaven, bere se jako výchozí informace ze vstupního seznamu. Velikost výsledné stránky se nezmění, pokud není explicitně uvedena.

Další užitečnou operací může být přerovnání stránek pro tisk knížky.

<code>book</code>	přerovná stránky pro tisk bookletu
<code>book nup(2)</code>	nejprve přerovná stránky pro tisk bookletu, pak umístí dvě stránky na jednu

Dále se mohou hodit následující operace.

<code>scalet(a4)</code>	zvětší/zmenší velikosti papíru na a4
<code>scalet2(1,2,3,4)</code>	zvětší/zmenší velikosti papíru na explicitně zadanou (levý dolní, pravý horní roh)
<code>paper(a4)</code>	nastaví velikosti papíru na a4 (neprovádí se zvětšení, zmenšení stránky)
<code>crop(a4)</code>	ořízne stránky na daný formát
<code>crop2(1,2,3,4)</code>	ořízne stránky na daný rozměr
<code>merge</code>	sloučí všechny stránky v seznamu na jednu
<code>cmarks</code>	přidá ořezové značky na stránku, jsou umístěny podle bboxu

Mezi zajímavé příkazy patří *modulo*, *select* a *apply*. U příkazu *modulo* udává první parametr *n* počet stránek ve skupinách, na které se rozdělí vstupní seznam. Pokud není celkový počet stránek

v seznamu násobkem čísla  $n$ , přidá se na konec seznamu příslušný počet prázdných stránek, aby seznam měl délku rovnu násobku čísla  $n$ . Pokud je nastaven parametr `round`,  $n$  je rovno  $\max(n, \text{round})$ . Na každou skupinu je aplikován výběr a provedou se příslušné příkazy. Pokud výběr stránky obsahuje symbol '-' vybírají se stránky ze skupiny, která je zrcadlově symetrická dle středu seznamu. Výsledná skupina seznamu je spojena do výstupního seznamu.

### Příklad:

Vezmeme seznam stránek '1 2 3 4 5' a aplikujeme na něj modulo s  $n$  rovným 3. Nejprve se seznam doplní na vhodnou délku '1 2 3 4 5 6' a rozdělí se na příslušné skupiny '1 2 3' a '4 5 6'. Vybereme {1 -1} '1 6' a '4 3' a výsledek spojíme '1 6 4 3'. Pokud parametr `half` je roven 1, zpracovává se pouze polovina skupiny. V tom případě je výsledek předchozího příkladu '1 6'.

### Příklady:

<code>modulo(2, half=1){-1 1 2 -2}</code>	odpovídá operátoru <code>book</code>
<code>modulo (1){-1}</code>	obrátil seznam stránek
<code>modulo(2){1 2 rotate(180)}</code>	odpovídá příkazu <code>duplex(1)</code> , tj. první stránka bez změny, druhá se rotuje o 180 stupňů
<code>modulo(16){1..16 book}</code>	vytvoření se signatura o 16-ti stránkách pro vázání knížek

Příkaz `apply` provede na vybraných stránkách zvolené příkazy. Příkaz `select` vybere stránky ze seznamu a případně na ně aplikuje příkazy.

<code>apply{1 new }</code>	za první stránku přidá prázdnou
<code>select{1..10}</code>	vybere prvních deset stránek z dokumentu

Příkaz `norm` sjednotí velikost všech stránek. Mezi jeho argumenty patří `center`, výchozí hodnota je centrování podle os  $x$  a  $y$ , `x` centrování podle osy  $x$ , `y` centrování podle osy  $y$ , `xy` centrování podle obou os a `none` vypne centrování.

Někdy mají stránky zbytečně velké okraje. V tomto případě je možné nechat GhostScript přepočítat jejich okraje. Tato operace se provádí pomocí příkazu `bbox`. Rozměry výsledného `bboxu` je maximum z `bboxu` jednotlivých stránek ze seznamu.

<code>bbox scaleto(a4)</code>	přepočítá okraj a přizpůsobí velikost stránek na rozměr <code>a4</code>
<code>bbox norm(center=y) scale(a4)</code>	přepočítá okraje a přizpůsobí velikosti stránek na papír <code>a4</code> , vhodné pro stránky se zrcadlově orientovanými okraji

V jistých případech je třeba očíslovat stránky, pro tento úkon je určen příkaz `number`.



<code>numer</code>	očísluje všechny stránky, začíná se číslem 1 a je umístěno dole na středu stránky
<code>number(10,10)</code>	obdobně jako předchozí případ, jen číslo je umístěno na pozici 10, 10 od levého dolního rohu
<code>number(start=2)</code>	bude se číslovat od 2
<code>number(text="-%d-")</code>	číslovat se bude od 1, ale výstup bude vypadat následovně "-číslo_stránky-" (" <code>%d</code> " říká, kde bude číslo stránky)

Další parametry jsou: `size` (velikost písma), `font` (název řezu písma, font musí být obsažen v příslušném formátu).

## Seznam a stručný popis všech příkazů

<code>apply{page_ranges}</code> <code>bbox</code>	aplikuje jednotlivé příkazy na výběr stránek přepočítá na každé stránce bounding box za pomoci GhostScriptu
<code>book</code> <code>cmarks(by_bbox)</code>	přerovná stránky do knížečky přidá na stránky ořezové značky, pokud je <code>by_box</code> roven 0, přidávají se dle rozměru stránky, v opačném případě dle bounding boxu
<code>crop(paper)</code> <code>crop2(lx, ly, hx, hy)</code> <code>del</code> <code>duplex(long-edge)</code>	ořízne stránku dle zadaných rozměrů papíru ořízne stránku dle zadaných rozměrů smaže všechny stránky v seznamu pootáčí stránky v seznamu pro duplexní tisk, parametr <code>long-edge</code> říká, podle jaké strany papíru se bude otáčet (1 delší, 0 kratší)
<code>flip(mode)</code>	zrcadlově otočí všechny stránky v seznamu, <i>horizontal</i>   <i>h</i> vodorovně, <i>vertical</i>   <i>v</i> svisle.
<code>line(b_x, b_y, e_x, e_y, size)</code>	nakreslí čáru šířky <i>size</i> mezi body <i>b_x</i> , <i>b_y</i> a <i>e_x</i> , <i>e_y</i>
<code>matrix(a,b,c,d,e,f)</code>	provede afinní transformaci souřadnic stránky dle zadané matice, význam jednotlivých argumentů viz. kapitola o afinní transformaci
<code>merge</code>	spojí stránky v seznamu do jedné (položí je přes sebe)
<code>modulo(pages, half, round){page_ranges}</code> <code>move(x, y)</code> <code>new{příkazy}</code>	posune obsah stránky o <i>x</i> , <i>y</i> vytvoří a přidá prázdnou stránku na konec seznamu a aplikuje na ní <i>příkazy</i>
<code>norm(center, scale)</code>	sjednotí velikost všech stránek v seznamu

<code>number(x,y,start,text,font,size)</code>	přidá na jednotlivé stránky čísla stránek, <i>start</i> počáteční číslo stránky, <i>font</i> jméno fontu, <i>text</i> , který se má na stránku vykreslit, výchozí „%d“ číslo stránky (podobný formátu printf), <i>size</i> velikost písma, všechny parametry jsou nepovinné
<code>nup(x, y, dx, dy, orientation, frames)</code>	přeskládá $x \times y$ stránek na jednu, $x$ počet stránek ve vodorovném směru, $y$ počet stránek ve svislém směru, $dx$ mezera mezi stránkami ve vodorovném směru, $dy$ mezera mezi stránkami ve svislém směru, <i>orientation</i> způsob rovnání stránek, <i>frames</i> stránky budou orámovány
<code>orient(landscape portrait)</code>	nastaví orientaci textu na stránkách v seznamu, neprovede se žádná změna obsahu stránky, tato informace je důležitá pro příkaz <code>nup</code> , ovlivňuje způsob kladení stránek
<code>paper(paper_size)</code>	nastaví velikost papíru stránkám v seznamu, neprovede se žádná změna měřítka
<code>paper2(x, y)</code> <code>read(name)</code>	nastaví explicitní rozměry papíru připojí na konec seznamu stránky ze souboru jménem <i>name</i>
<code>rotate(angle)</code> <code>scale(scale)</code>	otočí stránky o úhel <i>angle</i> ve stupních změní velikost stránek v zadaném měřítku $scale < 1$ zmenšení, $scale > 1$ zvětšení
<code>scaleto(paper, top, right, bottom, left)</code>	změní velikost všech stránek v seznamu, <i>paper_size</i> je velikost papíru, na kterou se stránka přetransformuje, okraje jsou explicitně nastaveny na půl palce
<code>scaleto2(x, y, top, right, bottom, left)</code>	změní velikost všech stránek v seznamu na explicitní velikost papíru, provádí se změna měřítka
<code>select{page_ranges}</code>	vybere stránky dle <i>page_ranges</i> , aplikuje na ně příslušné příkazy, výsledný seznam obsahuje pouze stránky z výběru
<code>text(x, y, text, size, font)</code>	nakreslí text na všechny stránky v seznamu, $x,y$ umístění na stránce, <i>size</i> velikost fontu, <i>font</i> jméno fontu, parametry <i>size</i> a <i>font</i> jsou nepovinné
<code>write(name)</code>	uloží seznam stránek do souboru jménem <i>name</i>

## Omezení

Soubory ve formátu PostScript nelze spojovat dohromady a jsou u nich vyžadovány DSC komentáře. V případě PDF formátu nelze pracovat s kryptovanými PDF. Program neumí pracovat s dokumentem PDF, který využívá některá rozšíření z verze 1.5 a novějších. Jedná se například o ObjectStreams a Cross-Reference Streams.

## Instalace

1. zprovoznit zlib a GhostScript<sup>2</sup>
2. získat zdrojové kódy ze svn,  
"svn co https://pspdftool.svn.sourceforge.net/svnroot/pspdftool pspdftool"  
nebo použít balíček se zdrojovými kódy a rozbalit jej  
"tar xvzf pspdftool.tgz"
3. přepnout se do adresáře se zdrojovými kódy  
"cd pspdftool/trunk"
4. konfigurace překlad a instalace  
"./configure && make && make install"

Program byl testován pod Linuxem, Mac OS Xem, Solarisem (sparc), HP-UXem a AIXem. Předpokládá se snadná přenositelnost i na ostatní UNIXové systémy, na kterých nebyl testován.

---

<sup>2</sup>zlib alespoň ve verzi 1.2.3, ghostscript alespoň ve verzi 8.60

# Kapitola 6

## Pspdftool - implemntace

V této kapitole se nachází popis jednotlivých částí pspdftoolu. Program je napsaný v jazyce C.

### 6.1 Zobecnění dokumentu

Dokument určený pro tisk obsahuje informace o stránkách a jejich vlastnostech. Pro reprezentaci stránek je zvolen spojový seznam. Jeho prvky odpovídají jednotlivým stránkám v dokumentu. Se seznamem lze jednoduše provádět následující operace: změny pořadí, duplikace, smazání a přeuspořádání stránek. Pro každou stránku v seznamu je uložena informace o rozměru papíru, velikosti bounding boxu, orientaci textu a část implementačně závislou na konkrétním formátu. Hlava seznamu obsahuje globální informace platné pro všechny stránky: maximální bbox, maximální velikost papíru, globální orientaci textu na stránce a implementačně závislou část na konkrétním formátu.

Knihovna je rozdělena na dvě části, část nezávislou na konkrétním formátu a část implementující konkrétní formát. Výhodou tohoto návrhu je snadná rozšířitelnost o další formát. Část nezávislá na formátu implementuje libovolné operace se seznamem, jako jsou přeuspořádání stránek, výběr stránek a podobně.

Část závislá na formátu implementuje následující funkce: načtení dokumentu, uložení dokumentu, spojení dvou dokumentů, transformaci jednotlivých stránek, spojení dvou stránek do jedné, kreslení a psaní textu na stránku, ořez stránky. Záměrem bylo navrhnout co možná nejjednodušší rozhraní mezi částí na formátu závislou a nezávislou. Například rotace nebo změna velikosti stránky se provede aplikací vhodné transformační matice na stránku.

Obecně nemusí být jednoduché implementovat všechny operace s konkrétním formátem (např. nelze jednoduše spojit dva různé PostScriptové soubory). Pro tento případ je implementován formát *dummydoc*, jehož interface je zavolán v případě absence určité funkce konkrétního formátu. Ve většině případů skončí toto volání chybou.

### 6.2 Část nezávislá na konkrétním formátu

Tato část knihovny obsahuje části nezávislé na konkrétním formátu. Patří sem interface pro práci a registraci konkrétních formátů, sada funkcí pro práci s transformační maticí, funkce pro práci se seznamem stránek a další pomocné funkce. Do této části můžeme řadit příkazový interpret, který popíše později.

## 6.3 Implementace konkrétních formátů

Pro ověření návrhu knihovny byly implementovány moduly pro následující formáty: PostScript a PDF. Tato implementace je také ukázkou, jak by se mohly přidávat další moduly.

Každý modul implementující musí obsahovat funkci, která je předána funkci *doc\_register\_format* pro přidání nového formátu do *pspdftoolu*. Při registraci modulu formátu je této funkci předána struktura definující interface mezi závislou a nezávislou částí na formátu. Funkce naplní ve struktuře ukazatele na rutiny pro práci s novým formátem. Struktura obsahuje jednoznačnou třípísmennou identifikaci formátu (z pravidla koncovka dokumentu).

Za zmínku stojí implementace funkce, která nastavuje stránce transformační matici. Pokud je funkce několikrát po sobě volána na stejnou stránku, nová matice se vynásobí s aktuální.

Ke každé stránce je přiřazeno počítadlo referencí, tím lze značně snížit spotřeba paměti. Při kopírování stránky se zvýší její počítadlo referencí, obsah stránky se nekopíruje. V případě změny jejího obsahu se stránka zkopíruje a původní stránce se sníží počítadlo referencí. Stránka se kopíruje pouze, pokud je potřeba ji změnit. Obdobný mechanismus je použit i pro kopírování seznamů.

### 6.3.1 PostScript

Od PostScriptového souboru je požadováno, aby obsahoval DSC komentáře. Bez nich nelze určit, aniž by bylo potřeba PostScript interpretovat, kde začíná a končí příslušná stránka dokumentu. PostScriptové soubory, jak bylo uvedeno výše, nelze jednoduše spojovat dohromady.

Nástroj PSUtils umožňuje spojovat pouze soubory se stejným prologem. Moje implementace pro PS formát neumožňuje spojovat žádné dokumenty.

Funkce pro načtení dokumentu funguje následovně: vstupní soubor se čte po řádcích a zpracovávají se řádky s DSC komentářem. Za pomoci těchto komentářů lze soubor rozdělit na následující části: hlavičku, *procdef*, *docsetup*, jednotlivé stránky a trailer. Každá část má nějaký význam, viz. kapitola o DSC komentářích.

Protože PS soubor může být velmi veliký (stovky MB), není mapován celý do paměti. U každé stránky se udržuje v paměti její offset v souboru.

Do sekce *procdef* se přidávají vlastní definice a redefinice PostScriptových operátorů potřebných pro práci knihovny. Například změna operátoru *showpage* na prázdný a místo něj definice nového.

Samotné operace typu transformace, kreslení čar, psaní textu jsou uskutečněny přidáním příslušných PS operátorů a jejich argumentů na začátek „programu“ stránky.

Spojení dvou stránek se provede spojením obsahů stránek, mezi jednotlivé stránky je vhodné přidat operátory pro uložení a obnovení grafického stavu interpretu PostScriptu. Na konec jednotlivých stránek se přidá nově definovaný operátor pro zobrazení stránky.

### 6.3.2 PDF

U PDF dokumentu je možné implementovat všechny v této práci zmíněné předtiskové úpravy. V dokumentu se přečtou všechny objekty. Objekty stránek se převedou na objekty XObject. K původní stránce se vytvoří nová stránka, která obsahuje XObject z původní stránky. U každé stránky v seznamu je nutné si zapamatovat příslušné číslo objektu, který ji reprezentuje. Při ukládání dokumentu se vygeneruje ze seznamu stránek strom stránek (dle specifikace PDF), odstraní se nepoužité objekty (prohledáním do hloubky od Catalog objektu). Do souboru se zapíše všechny použité objekty a na konec se přidá trailer a tabulka referencí.

Operace typu transformace, spojení dvou stránek do jedné, kreslení čar a psaní textů se týkají streamu příslušné stránky. Nové stránky jsou vytvářeny z odkazů na stávající XObjecty. Tyto

stránky nemají zakódovaný content stream. Není tedy potřeba implementovat jednotlivé filtry pro práci se streamem. Spojení dvou stránek odpovídá spojení dvou Content Streamů za sebe. Kreslení a transformace odpovídá přidání příslušných operátorů do Content Streamu stránky. Druhou výhodou daného řešení je jednoduché sloučení resources stránky.

Problém nastane, jestliže je potřeba zkonvertovat stránku, jejíž popis je rozdělen do více streamů. Přímočaré řešení by bylo zkonvertovat každou část na XObject. Vzhledem k tomu, že mohou být streamy mezi sebou závislé, nemusí být tento převod korektní ve všech případech. Tento problém lze vyřešit pouze sloučením více streamů do jednoho. Pro tento úkon je potřeba stream dekryptovat. Proto je program doplněn o funkce pro dekompresi streamu z LZW nebo deflate. Pokud nebude k dispozici filtr pro práci se streamem, program vygeneruje varování a použije výše citovaný nekorektní postup. Filtry jsou implementovány v souboru *pdf\_filters.c*.

Pro operaci spojení dvou dokumentů je potřeba v jednom z dokumentů přechíslovat objekty tak, aby nekolidovaly s druhým dokumentem. V *pspdftoolu* je použito následující řešení: v připojovaném dokumentu se přičte ke každému číslu objektu číslo udávající počet objektů v druhém souboru.

Při zápisu nového dokumentu se nejprve ze stránek v seznamu vygeneruje vyvážený strom a odstraní nepoužité objekty. Tato operace se provádí za pomoci prohledávání objektů do hloubky z catalog objektu. Po této operaci se zapíše objekty do souboru a vygeneruje se k nim cross reference table.

Implementace formátu plně podporuje PDF do verze 1.4. Novější PDF obsahují například tato rozšíření: ObjectStreams a Cross-Reference Streams. Protože tato rozšíření nejsou často využívána, nepovažoval jsem za důležité, aby s nimi uměl nástroj pracovat. K jejich implementaci by bylo vhodné umět pracovat se všemi filtry, které jsou definovány ve specifikaci PDF.

## 6.4 Interpret příkazů

V současné době je k dané knihovně napsán jediný uživatelský agent — jednoduchý interpret příkazů. Jeho implementace se nachází v souboru *cmdexec.c*. Tokenizér *cmd\_get\_token* rozkládá vstupní soubor na tokeny. Tokenizér je volán funkcí *cmd\_make\_tree*, jednoduchý LR1 parser, který z tokenů vytvoří derivační strom. Na výsledný derivační strom je zavolána funkce *cmd\_exec\_tree*, která vykoná zadané příkazy. Funkce *cmd\_exec* provede zadané příkazy na seznam stránek. Příkazy, které interpret obsahuje, jsou definovány ve struktuře *cmd\_commands*. První položkou je název příkazu, druhou jeho nápověda, třetí ukazatel na obslužnou funkci a poslední argumenty příkazu. Každé obslužné funkci je předán seznam stránek, pole argumentů funkce a selektor stránek<sup>1</sup>.

---

<sup>1</sup>viz syntaxe příkazu

# Kapitola 7

## Závěr

V této práci jsem prozkoumal možnosti předtiskových úprav dokumentů. Vytvořil jsem vlastní nástroj, který je komplexnější než řešení současná. Mezi jeho hlavní výhody patří: stejné rozhraní pro více formátů, možnost rozšířit o další formáty, případně konverze jednoho formátu na druhý s užitím externího programu.

Na tisku by se dalo dále zkoumat mnohé, například: techniky tisku, barevné kalibrace zařízení, nebo nástroje pro tvorbu dokumentů.

Mnou vytvořený nástroj `pspdftool` není primárně určen k profesionálnímu použití v tiskárnách. Neumí například pracovat s barvami, nebo přidat soutiskové značky. Zpravidla se v profesionálním prostředí vytváří dokument pro tisk přesně na míru, tj. není na něj potřeba aplikovat předtiskové úpravy. Dle mého názoru absence výše zmíněných operací není omezující při tisku černobílých dokumentů.

Dále by bylo možné doplnit k nástroji grafické rozhraní pro jeho intuitivnější ovládání. Příkazový interpret by se mohl rozšířit o definici uživatelských maker. Část pracující s PDF by mohla být rozšířena o další filtry a o další rozšíření novějších verzí PDF. Do části implementující práci s PostScriptem by bylo vhodné doplnit operace, které by například tiskárně řekly, jestli má používat duplex. Bylo by možné přidat do programu knihovny pro práci s ostatními formáty, například PCL nebo GDI. Další rozšíření této funkcionality by mohlo být zpracování PPD konkrétní tiskárny a na základě informací z PPD zapnout, nebo vypnout určitou předvolbu tisku.

Nový nástroj `pspdftool`, který je výsledkem této práce, je první obecný stránkově orientovaný nástroj pro práci s tiskovými formáty.

# Literatura

- [1] Adobe Systems, Inc., *PostScript Language Reference Manual*, 3rd edition, Addison-Wesley Professional, 1999.
- [2] Adobe Systems, Inc., *PDF reference Version 1.6, 5th edition*, Adobe Press 2004.
- [3] Adobe Systems, Inc., *PostScript Language Document Structuring Conventions Specification*, Adobe Press 1992.
- [4] <http://www.ghostscript.com/awki>
- [5] <http://www.tardis.ed.ac.uk/~ajcd/psutils/>
- [6] <http://www.accesspdf.com/pdftk/index.html>



# Dodatek A

## Obsah přiloženého CD-ROM

Na přiloženém CD-ROMu se nachází v adresáři *bc\_text* zdrojové kódy a text bakalářské práce, v adresáři *src* se nachází zdrojový kód programu, příklady jeho použití a jeho dokumentace v doxygenu.