

Zpravodaj Československého sdružení uživatelů TeXu

Vít Zýka

Používáme pdfTeX II: prezentace fotografií aneb jak na hypertext

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 12 (2002), No. 1, 13–21

Persistent URL: <http://dml.cz/dmlcz/149877>

Terms of use:

© Československé sdružení uživatelů TeXu, 2002

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ: The Czech Digital Mathematics Library* <http://dml.cz>

Používáme pdf_TE_X II: prezentace fotografií aneb jak na hypertext

VÍT ZÝKA

V prvním dílu [8] jsme se seznámili s vkládáním obrázků do pdf_TE_Xu. V tomto článku vytvoříme dokument obsahující naskenované fotografie a zamýšlený k projekci data-projektorem. Na této aplikaci si ukážeme tvorbu hypertextových odkazů, záložek a náhledů.

Prezentace fotografií

Také máte fotografie, které byste chtěli ukázat více lidem najednou? Řešením je fotografie naskenovat a pomocí nějakého vhodného softwaru je v celoobrazovkovém režimu promítnout data-projektorem. Je zřejmé, že taková projekce nedosáhne kvality diapozitivu (typické rozlišení data-projektoru je 1024 × 768 bodů), ale zase máme možnost obraz dodatečně upravit, třeba zvolit lepší kompoziční výřez nebo dorovnat kontrast či barevné podání. Podle mých zkušeností je na povídání o zážitcích z dovolené kvalita promítání uspokojivá.

K projekci se mi osvědčil Acrobat Reader. Umí celoobrazovkový režim, zobrazovat náhledy a záložky pro snazší orientaci, je rozšířený a zdarma. Také umožňuje pohyb po fotografiích pouze pomocí myši, což ocení zejména ti, kteří mají myš zabudovanou do dálkového ovladače projektoru, a nechtějí chodit ke klávesnici nebo hledat klávesy v zšeřelé místnosti. PDF dokument pro Reader vytvářím plain formátem pdf_TE_Xu.

Uživatelský soubor specifikující fotografie

Nejdříve načteme soubor maker, pak uvedeme popisek, který se vloží jako název kořenové záložky a celkový počet fotografií

```
1 \input vzdia.tex
2 \diaset{Altaj 2001}{26}
```

a dále, po specifikaci adresáře, základu a koncovky jmen souborů skenů, vkládáme jednotlivé fotografie. U každé uvedeme její popis do záložky a rozlišovací část jména souboru (první jméno fotky je `altaj01.jpg`).

```
3 \def\diapath{./foto/} \def\diabase{altaj} \def\diaext{.jpg}
4 \dia{Most přes Katuň}{01}
5 \dia{Belucha}{05}
```

```

6 % ...
7 \dia{Příjezd do Čemalu}{70} \end

```

Implementace maker v souboru vzdia.tex

Fotografie na celou stránku

Naskenované fotky vkládáme makrem `\dia`.

```

8 \def\dia#1#2{\tooutline{#1}{0}
9   \centerline{\putimage{\diapath\diabase#2\diaext}}\eject}

```

Název fotografie, který je jeho prvním parametrem, se na řádce 8 použije k vytvoření záložky. Jim se budeme věnovat na straně 16. Fotografie umístíme do vycentrovaneho řádku a odstránkujeme pomocí `\eject`. V makru `\putimage` zajistíme, aby fotka na stránce zaujímala co největší rozměr.

```

10 \def\putimage#1{\pdfximage height\pdfpageheight{#1}%
11   \setbox0=\hbox{\pdfrefximage\pdflastximage}%
12   \ifdim\wd0>\pdfpagewidth\pdfximage width\pdfpagewidth{#1}%
13   \else\pdfximage height\pdfpageheight{#1}\fi%
14   \hbox{\pdfrefximage\pdflastximage}}

```

Na řádce 10 ji cvičně načteme do boxu tak, aby byla vysoká přesně na výšku stránky. Protože pdf_TE_X sám proporcionálně dopočítá i šířku, můžeme porovnat šířku tohoto boxu s šířkou stránky a podle toho ji definitivně vložit s největším možným zvětšením.

V minulého dílu [8] jsem omylem uvedl starší syntaxi parametrů primitivu `\pdfximage`. To sice platí, ale dnes můžeme použít přímější zápis atributů objektu, včetně geometrické transformace:

```
\pdfximage [(rule spec)] [attr{(attributes)}] [page n] file{filename}
```

Takto můžeme do nějakého jiného dokumentu vložit druhou stranu tohoto článku zmenšenou na šířku 6 centimetrů: `\pdfximage width6cm attr{/Matrix [0 1 -1 0 595.3 0] /BBox [0 160 50 0] } page 2 {dia-bul.pdf}`. A co se se stránkou ještě stalo? Byla oříznuta 160 bp zespodu a 50 bp zprava, dále otočena doleva o 90° a posunuta o 21 cm doprava.

Stránky nebudou obsahovat záhlaví ani zápatí. Dále chceme, aby fotografie dosahovala až k okraji stránky a byla vertikálně vycentrována. Proto předefinujeme výstupní rutinu, aby nevkládala dodatečné okraje. Nebojte se, bude velmi jednoduchá.

```

15 \output={\shipout\diabox \advancepageno}
16 \def\diabox{\vbox to\vsizel{
17   \vbox to0pt{\hrule width\hsize height\vsizel depth0pt\vss}
18   \vfil\unvbox255 \vfil}}

```

Trochu vysvětlení si zaslouží řádek 17. Do výstupního vertikálního boxu vkládá linku o velikosti celé stránky zajišťující černé pozadí fotografie. Je uvnitř boxu nulové výšky, takže neposune bod sazby.

Odkazy (links and destinations)

Hypertextový odkaz může být externí (mimo dokument) a interní. Externí odkaz je zadáván pomocí URL (Unique Resource Locator), interní tvoří nejčastěji dvojice odkaz–doskok. V tomto případě jsou obě složky provázány jednoznačným identifikátorem v podobě čísla nebo jména.

Doskok

Doskok se definuje pdfTeX primitivem

```
\pdfdest <num n | name{refname}> <dest spec>
```

Doskok pojmenovaný `PokusnyDoskok` zachovávající aktuální zvětšení dokumentu zadefinujeme takto

```
\pdfdest name{PokusnyDoskok} xyz
```

V naší aplikaci budeme potřebovat doskok na každou stránku s fotografií pro správnou funkci záložek. Musíme tedy vytvořit jméno jednoznačné pro každou stránku. Použijeme-li číslo stránky (je uložen v nultém registru), může vypadat takto

```
\pdfdest name{page:\the\count0} fit
```

Protože vždy budeme chtít zobrazit celou fotografii v největší velikosti, která se vejde do aktuálního okna, použili jsme místo `xyz` přepínač `fit` [7].

Někdy se hodí odkázat část dokumentu ve zvoleném zvětšení. To umožňuje doskok se syntaxí `\pdfdest name{CtyrikrazZvetsi} xyz zoom 4000`. Referenční bod sazby není vložením doskoku nijak ovlivněn, ale je zaznamenána jeho poloha. Při doskoku má prohlížeč nastavit tento bod do levého horního rohu okna a zvětšit/zmenšit podle faktoru `zoom/1000`.¹

Protože budeme vždy zobrazovat jednu fotografii na celou stranu, můžeme doskok umístit do kteréhokoliv místa na stránce, třeba před nebo za fotografií. Lepší však bude vložit jej přímo do výstupní rutiny. V ní bude zaručeno, že na každou stranu bude vložen právě jeden doskok bez ohledu, zda dojde k explicitnímu nebo automatickému zlomu strany². Za řádek 16 přidáme

```
19 \pdfdest name{page:\the\count0} fit
```

Odkaz

Odkaz vytvoříme dvojicí primitivů

```
\pdfannotlink [<rule spec>] [attr{<attributes>}] <action spec>
```

¹Prohlížeč `xpdf v0.91` parametr `zoom` zcela ignoruje. Acrobat naopak přidává další, vcelku užitečné akce. Je-li dokument v požadovaném zvětšení menší než okno prohlížeče, je vycentrován; je-li referenční bod příliš u okraje, takže by zobrazená část dokumentu nevyužívala celou plochu okna, náležitě jej posune.

²Explicitní stránkový zlom je vyvolán vložením `\penalty=-10000` ve vertikálním módu. Tato penalta je obsažena v plainovém makru `\eject` nebo L^AT_EXovém `\newpage`. Automatický zlom strany provede T_EX sám na základě optimalizačního algoritmu plnění strany [3, str. 238].

a `\pdfendlink`. Aktivní plocha linku je vypočítána z obsahu mezi oběma primitivami. Uvnitř může dojít k řádkovému i stránkovému zlomu, ale musí být ve stejné úrovni boxu nebo skupiny. Jinou možností je specifikovat aktivní plochu pomocí `<rule specification>`. Makro, které vytvoří odkaz na jmenný doskok uvedený v jeho prvním parametru s aktivní plochou textu v druhém parametru, může vypadat takto

```
20 \long\def\hlink#1#2{\pdfstartlink attr{/Border [0 0 0] /H/N }
21 goto name{#1}#2\pdfendlink}
```

Uvedenými atributy požadujeme, aby prohlížeč nekreslil okolo linku žádný rámeček a neprovedl žádné zvýraznění při kliknutí (`highlighting=none`). Celkový výčet možných atributů lze najít v referenčním manuálu [6].

Chtějme v naší aplikaci umožnit kliknutím skok z každé stránky na stránku předcházející. Aktivní plochou takového linku nechť je levá třetina stránky. Nastavené atributy zajistí ‚neviditelnost‘ odkazu, takže nebude rušit při prohlížení fotek. Odkaz znovu umístíme do výstupní rutiny, za řádek 17.

```
22 \count1=\count0 \advance\count1 by-1
23 \vbox to0pt{\hbox{\hlink{page:\the\count1}}{%-
24 \vbox to\vsize{\hrule width.3\hsize height0pt depth0pt
25 \vss}}}\vss}
```

I zde je vše vloženo do boxu o nulových rozměrech, aby se neposunul bod sazby. Na řádce 24 vytvoříme box o rozměrech třetiny stránky, a ten vložíme jako druhý parametr makra `\hlink`. Jmenný odkaz musíme vytvořit stejně, jako pro doskok na řádce 19, jen číslo aktuální stránky zmenšíme o jednu (řádek 22).

Jinou možností, jak přejít na předchozí stránku, je využít tzv. *menu akce* (uživatelsky pojmenované akce). Tento název byl zvolen proto, že tímto způsobem jsou dosažitelné vesměs ty operace, které umožňuje Acrobat ve svých menu. Tedy například `NextPage`, `PrevPage`, `FirstPage`, `LastPage`.³ Jejich použití umožňuje makro podobné `\hlink`, jen prvním parametrem bude jméno akce.

```
\long\def\menulink#1#2{\pdfstartlink attr{/Border [0 0 0] /H /N}%
user{/Subtype /Link /A << /S /Named /N /#1 >> }#2\pdfendlink}
```

Záložky (outlines)

Záložky tvoří jakýsi aktivní obsah usnadňující orientaci v dokumentu. Dobrý prohlížeč je umí zobrazit paralelně s textem. Každá záložka obsahuje zobrazený text, akci (většinou specifikaci doskoku) a počet jejích vnořených záložek. Tento počet určuje hierarchickou strukturu spolu s pořadím vložení záložek.

Vraťme se k našemu příkladu. Na řádce 2 jsme uvedli údaje pro vytvoření kořenové záložky. Kód makra `\diaset` je totožný s kódem `\tooutline`, které záložky vytváří a jehož vysvětlení dlužíme čtenáři od prvního použití v makru

³Mezi další akce uvedme `GeneralInfo`, `Print` nebo `GoToPage`. Ale pozor! Tyto další pojmenované akce způsobí, že dokument nebude podle referenčního manuálu ‚portable‘. Krom toho je podporuje až Acrobat verze 3 a výše.

`\dia` na řádku 8. Synonymum vytvoříme jen proto, abychom nemátli uživatele.

```
28 \let\diaset=\tooutline
    Definice
29 \input il2ascii.tex
30 \def\tooutline#1#2{\iltwoascii{#1}%
31 \pdfoutline goto name{page:\the\count0} count #2{\ascii}}
```

je založena na primitivu

```
\pdfoutline <action spec> [count n] {{general text}}
```

Celé číslo $|n|$ označuje počet přímých podzáložek. Je-li $n < 0$, pak v počátečním stavu budou podzáložky svinuté. Vynecháme-li parametr `count n`, bude se považovat za nulový. V našem příkladě bude po úvodní titulní záložce odkazující na první fotografii dalších 26 rozvinutých podzáložek. Pro každou ze šestadvaceti fotografií ji přidá makro `\dia`.

Dále jsme na řádku 29 načetli makra pro převod znaků v kódování IL2 na sedmibitové nehackované ASCII. To souvisí s tím, že Acrobat záložky zobrazuje pouze systémovým fontem v kódování IL1 nebo UNICODE.⁴ Makro `\iltwoascii` zdefiniuje `\ascii` obsahující překódovaný text.

Chceme-li mít záložky správně česky nebo slovensky, překódujeme vstup do UNICODE. Inspiraci i funkční prototyp tohoto postupu z kódování IL2 nabídl Jirka Osoba v odpovědi v konferenci C_T_EXu `gopher://cs.felk.cvut.cz:70/OR63946-69054-gopher_root%3A%5B_lists.cstex%5Dcstex.2002-02%3B1`. Použijeme je pak takto:

```
32 \input 88592-unicode.tex
33 \def\tooutline#1#2{\toUNICODE{#1}
34 \pdfoutline goto name{page:\the\count0} count #2{\UNICODE}}
```

Náhledy stránek (thumbnails)

Pro vytvoření miniaturních náhledů stránek vytvořil Heiko Oberdiek balíček `thumbpdf`. Funguje pro plain T_EX i L^AT_EX a do zdrojového souboru pdfT_EXu stačí vložit

```
35 \input thumbpdf.sty
```

přeložit T_EXem, pak spustit skript v Perlu `thumbpdf tex_file_without_ext` a na závěr ještě jednou přeT_EXovat. Kromě Perlu je potřeba mít ještě nainstalován GhostScript.

Acrobat (Reader) umožňuje vyvolat panel s náhledy, a to i v celoobrazkovém režimu.⁵ Tato funkce se s úspěchem využije při hledání vhodného slajdu při prezentacích. Tlačítko vyvolávající tento panel lze ukázat pomocí

```
\menulink{ShowThumbs}{Náhledy}
```

⁴Tento šestnáctibitový font však nemusí být vůbec nainstalován, jako na mém Acrobat Readeru v4 pro Linux. Pro přenositelnost dokumentu je tedy sedmibitová ASCII vhodnější, byť přijdeme o diakritiku.

⁵Bohužel Acrobat Reader v4 pro Linux neumí panel s náhledy zavřít, čímž je tato funkce pod Linuxem nepoužitelná; `xpdf` ani `gv+gs`, pokud vím, náhledy ani záložky v současné době nezobrazují.

Rozměry a další parametry

Tím jsme nadefinovali všechna makra našeho balíčku pro prezentaci fotografií. Ještě zbývá nastavit několik parametrů, hlavně velikost stránky.

```
37 \pdfpageheight=210 truept \vsize=\pdfpageheight
38 \pdfpagewidth =280 truept \hsize=\pdfpagewidth
39 \hoffset=-1in \voffset=-1in
40 \parindent=0pt
41 \offinterlineskip
```

Výška byla nastavena na výšku nejběžnějšího tiskového formátu A4 – to kdybychom chtěli nějakou fotku tisknout. I tak však budeme muset zapnout parametr tisku `Fit to Page`. Šířku papíru jsme zadali tak, aby poměr stran byl 3:4, tak jak je u monitorů a projektorů zvykem.

Zřetězení článku (*article threads*)

Ještě si popíšeme použití hypertextového objektu, které autoři lokalizace Acrobatu nazývají *zřetězením článku*. S tématem tohoto příspěvku souvisí, protože umožňuje rychlejší navigaci ve složitějším dokumentu. Je-li třeba text v několika sloupcích a vložíme-li každý sloupec do takového vlákna, pak prohlížeč (Reader) kliknutím skočí do prvního sloupce, roztáhne jej na šířku okna, dalším klikáním jej bude vertikálně posouvat, dokud nenarazí na jeho dolní konec a pak zobrazí začátek dalšího sloupce. Dospějeme-li ke konci posledního sloupce, který jsme do vlákna vložili, zobrazí se zase stránka tak, jak byla před vstupem do vlákna. Jiným příkladem použití může být skupinová fotografie, kterou nejdříve chceme zobrazit celou a pak postupně ukázat detaily každého jedince. S pomocí zřetězeného článku se jedním kliknutím posuneme do dalšího výřezu.

Mohlo by se zdát, že tento mechanismus lze nahradit soustavou obyčejných odkazů a doskoků. Vidím zde však dva hlavní rozdíly:

1. Zřetězené články se mohou překrývat. Vstoupíme-li do nějakého stavu článku, pak je aktivní celá plocha okna pro posun dále (se shiftem zpět) v článku. Plochy ostatních článků v dokumentu nebo jiných stavů tohoto článku jsou momentálně neaktivní.
2. Nevejde-li se vertikálně celý blok textu zřetězeného článku do okna, zobrazovač jím (na uživatelské kliknutí) pohybuje, dokud jej celý nezobrazí, a teprve pak skočí do dalšího stavu článku.

Na vytvoření jednoho stavu článku slouží primitiv

```
\pdfthread [<rule spec>] [attr{<attributes>}]  
  <num n | name{<refname>}
```

nebo dvojice `\pdfstartthread \pdfendthread`. Syntaxe `\pdfstartthread` je stejná jako u `\pdfthread`. Dvojice značek se musí objevit ve vboxu stejné úrovně. Obsah pak stanoví velikost aktivní oblasti stavu článku. U jednoduché značky

se velikost dopočítá (není-li udána pomocí `<rule spec>`) podle rodičovského boxu. Velikost oblasti se dále zvětší o hodnotu registru `\pdfthreadmargin`.

Jak na to v L^AT_EXu?

Zde uvádím přehled popsaných možností pdfT_EXu pro uživatele L^AT_EXu. Spočívá v načtení patřičného makrobalíku a použití uživatelsky jednoduchého makra. Nejprve splatíme dluh z minulého dílu.

Vkládání obrázků

```
\usepackage[pdftex]{graphicx}
\includegraphics{file}
```

Pokud jméno souboru neuvedeme s příponou, bude se defaultně hledat `.pdf`, je-li při vložení balíčku `graphicx` uveden parametr `pdftex`. Bez tohoto parametru se bude hledat `.eps`. Chceme-li, aby náš dokument šel přeložit jak pdfL^AT_EXem, tak obyčejným L^AT_EXem, použijeme

```
\usepackage{ifpdf}
\ifpdf \usepackage[pdftex]{graphicx}
\else\usepackage{graphicx}\fi
```

a neuvedeme příponu jména souboru; pdfT_EX navíc bude umět načíst i soubory typu `.png`, `.jpg` a `.tif`. Obrázku z METAPOSTu musíme dát koncovku `.mps` (nebo udělat link např. `ln -s mpfile.1 mpfile1.mps`) a s touto koncovkou jej načíst.

Balík `graphicx` hledá obrázky ve stejných adresářích jako L^AT_EX makrobalíky⁶ plus v adresářích, které nastavíme `\graphicspath{{fig/},{../photo/}}`. Doporučená literatura: [5, 1].

Hypertext

```
\usepackage{hyperref}
\hypertarget{label}{}
\hyperlink{label}{contents}
\href{url}{contents}
```

Chceme-li, aby link nevytvářel okolo `contents` rámeček, nebarvil jej a nezpůsobil jeho inverzi při kliknutí, pak použijeme následující makro `\shyperlink`

```
\def\noBorderLinkAttr{\makeatletter%
\def\pdfBorderAttrs{/Border [0 0 0] /H /N}%
\def\hy@colorlink##1{\begingroup}
```

⁶Ve Web2C instalaci (T_EXLive) viz řádek `TEXINPUTS.pdflatex=...` v souboru `texmf.cnf`.


```

\let\Hy@colorlink=\hy@colorlink
\def\@pdfhighlight{/N } \def\@pdfborder{0 0 0}
\def\@urlcolor{black}%
\makeatother}
\long\def\shyperlink#1#2{{\noBorderLinkAttr\hyperlink{#1}{#2}}}
\long\def\shref#1#2{{\noBorderLinkAttr\href{#1}{#2}}}

```

Doporučená literatura: [2, 4].

Závěr

Budeme-li chtít rozdělit fotografie do několika skupin – třeba fotografie Filipa, Hanky a Milana, a dále fotografie z pobytu v horách a fotografie ze splutí řeky – můžeme přidat do záložek podtitulky s vlastní rozbalovací skupinou záložek. Protože se tak některé fotky mohou vyskytnout v několika skupinách zároveň, vyplátí se použít referování jednou použitých fotografií, jak bylo popsáno v minulém čísle Zpravodaje. Dokument tak nebude zbytečně narůstat tím, že jeden obrázek je zároveň na několika stránkách.

Na jednoduché aplikaci jsme ukázali základní vytváření hypertextových odkazů, záložek a náhledů. Kód maker je přístupný na adrese <ftp://cmp.felk.cvut.cz/pub/cmp/users/zyka/dia>.

Odkazy

- [1] David P. Carlisle. *Package in the ‘graphics’ bundle*, 1999. TeXLive6: `\$TEXMF/doc/latex/graphics/grfguide.ps.gz`.
- [2] Heiko Oberdiek. *PDF information and navigation elements with hyperref, pdfTeX, and thumbpdf*, 1999. TeXLive6: `\$TEXMF/doc/latex/hyperref/slides.pdf`.
- [3] Petr Olšák. *TeXbook naruby*. Konvoj Brno, 1. edition, 1997. Czech.
- [4] Sebastian Rahtz. *Hypertext marks in L^AT_EX: the hyperref package*, 1998. TeXLive6: `\$TEXMF/doc/latex/hyperref/manual.pdf`.
- [5] Keith Reckdahl. *Using Imported Graphics in L^AT_EX 2_ε*, 1997. TeXLive6: `\$TEXMF/doc/guides/epslatex/epslatex.ps.gz`.
- [6] Adobe systems Incorporated. *PDF reference manual, v. 1.4*, second edition, 2000. <http://partners.adobe.com/asn/developer/acrosdk/DOCS/PDFRef.pdf>.
- [7] Hàn Thê Thành, Sebastian Rahtz, and Hans Hagen. *The pdfTeX user manual*. Pragma, 2001. <http://www.tug.org/applications/pdftex/pdftex-s.pdf>.
- [8] Vít Zýka. Používáme pdfTeX: vkládání obrázků. *Zpravodaj Československého sdružení uživatelů T_EXu*, 11(4):181–186, December 2001.

Summary: Using pdfTeX II: document for photographs presentation; how to create a hypertext

This article describes making hypertext objects by pdfTeX. We focus on the hyperlinks, bookmarks, thumbnails, and article threads. The usage is demonstrated on a simple application—the document for photographs projection. We show creating hypertext on both the pdfTeX primitive level and with standard L^AT_EX macro-packages. This text follows the last issue article *Using pdfTeX: graphics inclusion*, C_STUG Zpravodaj 4/2001.

Vít Zýka, zyka@cmp.felk.cvut.cz

Textové mezery v T_EXu

JOSEF TKADLEC

S tím, jak roste kvalita tisku dostupného běžnému uživateli, roste i význam respektování typografických pravidel. Máme-li k dispozici kvalitní fonty, pak i „drobnosti“ mohou významně přispět k přiblížení se typografickému ideálu: k sazbě, která v hrubém pohledu vypadá jednoduše šedivě (takže při četbě neruší nepatřičnými bílými plochami) a přitom jsou zřetelně oddělena jednotlivá slova či logické celky. Někomu připadají některá typografická pravidla jako zbytečná nebo jen jako „pouhá“ estetická záležitost. Myslím si, že valná většina má i ryze praktický význam – umožňují, aby se co nejvíce snížilo čtenářovo úsilí potřebné k „dekódování“ napsaného textu.

V tomto článku se budu zabývat *českými* typografickými pravidly psaní mezer, některé národní typografie mají pravidla odlišná. Například v českých typografických pravidlech se upřednostňuje rovnoměrná šedivost stránky, zatímco jinde spíše vyznačování logických celků a grafické znázornění pomlek při čtení – to se projevuje například ve volbě mezery za tečkou na konci věty (zúžená nebo naopak rozšířená). Nemyslím si, že by česká pravidla byla ta nejlepší a jiná méněcenná, oba přístupy mají svá pro a proti.

Typografická pravidla ve starší literatuře [1, 3] vycházejí z omezených možností ruční sazby (hodně diskrétní změny velikostí) a jsou uváděna spíše na příkladech. Pro počítačovou sazbu jsou jednak zbytečně hrubá, jednak nevhodně formulovaná. Novější literatura se zase zabývá spíše některým konkrétním produktem, vychází z mezer, které má k dispozici, a dost často se vyskytují obraty