

Zpravodaj Československého sdružení uživatelů TeXu

Alan Hoenig

Značkování obrázků v dokumentech TeXu

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 1 (1991), No. 2, 18–24

Persistent URL: <http://dml.cz/dmlcz/148778>

Terms of use:

© Československé sdružení uživatelů TeXu, 1991

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>

```

\hskip 1cm{${\rightleftharpoons$}\hskip 1cm
\structure{\phantom{H}\swphantom\atom{N}\side{\sseabove\atom{H}}
\side{\sebelow\atom{H}}\nnesingle\atom{C}\side{\nnwabove\atom{H}}
\side{\nwbelow\atom{H}}\esingle\atom{C}\side{\nnedouble\atom{O}}
\ssesingle\atom{O}\esingle\atom{H}}
}}

```

(*Zbyněk Linhart*)

Krátký příspěvek Zbynka Linharta by měl povzbudit naše kolegy, kteří se zabývají sazbou chemických textů. Uvítali bychom z jejich praxe podrobnější informaci o tom, co existuje v oblasti veřejně dostupných programů, co se dá koupit, a pod.

(*-ov,-, -ju-*)

Značkování obrázků v dokumentech \TeX u

Alan Hoenig, John Jay College/City University of New York
 Email: ajhjj@cunyv

Přeloženo z *TUGboat* 12(1), 1991, str. 125–128.

Úvod

Problém včleňování obrázků do \TeX ovských dokumentů, který se dotýká řady lidí, ve skutečnosti už neexistuje. Díky příkazu `\special` je možné do tištěných dokumentů zařadit rozmanité druhy grafiky. Obecný postup je následující.

Požádejte \TeX , aby pro obrázek uvolnil v dokumentu bílé místo vhodných rozměrů. Připravte nějakým jiným programem samostatný soubor obsahující instrukce pro vytvoření obrázku. „Vtáhněte“ tento samostatný soubor do *dvi* souboru pomocí příkazu `\special`. Nakonec použijte ovladač, který přijímá vnější grafické soubory. Takových ovladačů již existuje mnoho.

Tento přístup však má i své nevýhody. Především grafický soubor není doopravdy součástí zdrojového souboru \TeX u. To znamená, že nebude možné prohlížet tento soubor na obrazovce, nepoužíváte-li ovšem zvláště chytrý obrazovkový ovladač (takový, který reaguje na příkazy `\special` a je natolik inteligentní, že ví, jak tento materiál zobrazit).

Potřeba značek v obrázku

Jiná nevýhoda se ukáže, jakmile je potřeba vpisovat nějaké značky přímo do obrázku. Jak je tam vložit, aby byly správně umístěny? Někteří přidávají značky přímo jako součást grafiky, což ovšem vede k vizuální nesourodosti, poněvadž font použitý pro tyto značky nejspíš nepatří do rodiny *Computer Modern*, k níž inklinuje \TeX . V ideálním případě bychom chtěli, aby značky sázel sám \TeX tak pěkně, jako celý zbytek dokumentu.

V tomto článku bych rád popsal jeden možný přístup ke značkování, který se mi dobře osvědčil. S použitím METAFONTu vytvořím vlastní obrázek v podobě fonu. Přitom se průběžně rozhoduji, které body v obrázku budou značkovány (avšak nemusím určovat značky samotné), a METAFONT zaznamená souřadnice těchto bodů do fonu. Potom, když \TeX včleňuje obrázek do textu, může tyto souřadnice použít ke stanovení, o kolik je potřeba box s obrázkem posunout doprava či doleva a nahoru či dolů.

Další práce v této oblasti

Na tomto poli se toho zřejmě zatím moc neudělalo. Rick Simpson byl nejspíš prvním, kdo (přínejmenším písemně) navrhl použít jako nástroj pro kreslení METAFONT. Jeho článek [2] stojí za přečtení.

Dalším příspěvkem je pak balík. maker Metaplot, vytvořený Patricií Wilcoxovou [3]. Patricia ukázala, jak je možné pojmout rozhraní mezi uživatelem a METAFONTem. Použila k tomu jazyk jednoho z běžně používaných souřadnicových zapisovačů. Jeho různé příkazy byly přeloženy do příkazů METAFONTu. Vznikl tak balík: s nímž je možné tvořit důmyslnou i pěknou grafiku. (Patricia je jedinec pozhnaný značně nadprůměrnými uměleckými dovednostmi.) Tyto možnosti jsou bohatě ilustrovány ve výše zmíněném článku. (Některé z mých strategií byly inspirovány návrhy Patricie Wilcoxové a Toma Rokickiho, jimž tímto s potěšením děkuji za jejich bezděčnou pomoc.)

John Hobby [1] volí jiný přístup. Upravil METAFONT tak, že dělá výstup přímo v PostScriptu namísto *generic font* formátu, který METAFONT normálně produkuje. I když by bylo možné generovat *Computer Modern* fonty ve formátu PostScriptu, je tento program *MetaPost* především prostředkem pro tvorbu obrázků a emblémů. Pro včleňování značek do obrázku užívá jeho program odlišný přístup.

Úloha METAFONTu

Nejprve se podíváme na to, jak pracuje METAFONTovská část tohoto projektu. Dejme tomu že každý obrázek je znakem „A“ ve fonu vytvořeném METAFONTem pro tento zvláštní účel a že tento font neobsahuje žádné jiné znaky. \TeX nikterak nevyžaduje, že „A“ musí vypadat jako opravdové „A“ a nemá žádné apriorní představy o správné velikosti takového znaku, takže si nikdo nebude stěžovat, pokud METAFONT požádáme o nakreslení „A“, které nebude vypadat vůbec jako „A“, ale jako obrázek, který potřebujeme v dokumentu. Předpokládáme, že zdrojový soubor METAFONTu, obsahující instrukce pro nakreslení obrázku, se jmenuje **figfont.mf**. Potom, za předpokladu, že jste všechny věci spojené s METAFONTem dobře zvládli, můžete vysadit dotyčný obrázek po deklaraci fonu

```
\font\figfont=figfont
```

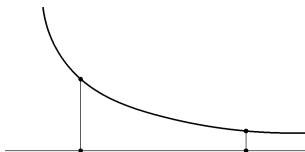
zadáním příkazu

```
{\figfont A}
```

na místě, kde se má obrázek objevit. (Nezapomeňte na uzavření do složených závorek!)

Následuje obrázek, který by se mohl dobře objevit v nějakém matematickém textu. (Ve skutečnosti se vyskytuje kdesi v prvním svazku Knuthova díla *Art of*

Computer Programming. Křivka reprezentuje část grafu funkce $f(x) = 1/x$. (Proto se zdrojový soubor pro METAFONT jmenuje `1onx.mf`.)



Bude užitečné probrat instrukce METAFONTu, jimiž se tento obrázek stvoří.

```
mode_setup;
u#=12pt; nib#=.5pt;
if (mode=smoke) or (mode=proof):
u#=.5pt#; nib#=.1pt; fi
define_pixels(u, nib);
beginchar("A", 12u#, 6u#, 0);
pickup pencircle scaled 2nib;
z1=(w/8,h); z2=(w/4, h/2);
z3=(w/2, h/4); z4=(w,h/8);
  % points on the curve
path p; p=z1..z2..z3..z4;
draw p; % draw the $1/x$ curve
z.x= point 2.6 of p; % another point
z5=(x2,0); z6=(x.x,0);
  % points on the $$-axis
pickup pencircle scaled nib;
draw origin--(w,0); % bottom axis
draw z2--z5; draw z.x--z6;
  % vertical struts
pickup pencircle scaled 6nib;
drawdot z2; drawdot z.x;
drawdot z5; drawdot z6;
endchar; bye.
```

Tyto příkazy jsou ve své většině rozumně samovysvětlující, jste-li ovšem seznámeni se základy syntaxe METAFONTu. Proměnná `nib#` udává průměr kreslicího pera a `u#` je *ad hoc* zvolená jednotka délky, která je vhodná s ohledem na celkové měřítko obrázku. Z dalších proměnných tohoto programu jsou `w` a `h` šířka a výška obrázku, a veličiny `z1`, `z2`, atd. (včetně `z.x`) se vztahují k význačným bodům v obrázku.

Rádky

```
u#=12pt; nib#=.5pt;
if (mode=smoke) or (mode=proof):
  u#=.5pt#; nib#=.1pt; fi
```

asi popudí skalního METAFONTistu; plyne z nich, že `u#` a `nib#` nabývají hodnot závislých na zařízení, což je v rozporu s duchem METAFONTu. Tato část zdrojového textu je diktována omezením mého monitoru. METAFONT je zvyklý pracovat se znaky o velikosti (zhruba) deset na deset tiskařských bodů. Během korektur na obrazovce (když mode je buď `smoke` nebo `proof`) používá METAFONT celou obrazovku pro zobrazení znaku. Je-li charakteristická délka znaku větší než několik jednotek pc, zobrazí METAFONT jen část písmena, pokud nezvolíme `u#` tak malé, aby bylo na monitoru místo pro zobrazení celého znaku. Výše uvedené řádky se pokoušejí implementovat tuto strategii. Charakteristické rozměry `u#` a `nib#` mají velikost, kterou bychom chtěli pro definitivní vytvoření fontu. Během korektur jsou však mnohem menší, aby umožnily prohlížet dílo na obrazovce počítače.

Komunikace METAFONTu s T_EXem

T_EX bychom mohli nyní použít pro popisný text k obrázku, víc bychom však obrázku pomohli připojením značek k některým jeho bodům. Knuth doplnil ke každé z okrouhlých teček značku—můžeme to udělat také?

Počítačová experti obvykle charakterizují T_EX a METAFONT jako plnohodnotné programovací jazyky, METAFONT ovšem tuto roli tak docela neplní. S výjimkou souborů `log` a `gf` totiž nemá jiné výstupní možnosti. Proto nemůžeme zapsat souřadnice značených bodů do souborů a předat je v této formě T_EXu.

Klíčovým bodem je rozpoznání užitečnosti parametru `fontdimen`. METAFONT normálně používá tyto parametry k zaznamenání univerzálních konstant pro určitý font, jakou je třeba šířka výplně *quad* nebo odstup mezi slovy. Nezdá se, že by byla nějaká horní mez počtu dovolených parametrů `fontdimen`, ani omezení způsobu jejich aplikace. Proto jsem si dovolil použít je k uschování souřadnic každého značeného bodu. Každý bod zabírá dva parametry `fontdimen`—jeden pro souřadnici *x* a druhý pro souřadnici *y*.

Soubor `convert.mf` obsahuje definice maker jako `convertz_`, jež transformuje souřadnice do použitelného formátu. Hned za `mode_setup` je tedy třeba uvést příkaz

```
input convert;
```

Před instrukcí `endchar` pak přidáme další řádek

```
fontdimen10: convertz_(z2,z.x,z5,z6);
```

Makro `convertz_` převádí seznam párů na čísla, která odpovídají syntaxi pro `fontdimen`.

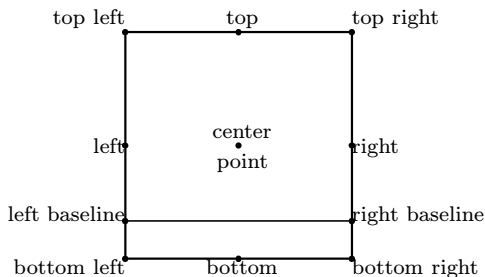
Úloha T_EXu

Jak již bylo řečeno, „zMETAFONTované“ obrázky lze snadno včlenit do T_EXovského dokumentu. Je však také snadné připojit značky. Pro toto použijeme sadu maker, jejichž obdoba je popsána v T_EXbooku, Appendix D. Jsou-li `\x` a `\y` T_EXovské *dimen* registry, můžeme je jednoduše naplnit hodnotami z vhodného parametru `\fontdimen`:

```
\advance\fontdimencount by 1
\x=\expandafter \the\fontdimen
\the\fontdimencount \figfont
```

(a podobně pro $\backslash y$). Poté, co do boxu $\backslash labelbox$ vložíme text pro značku, konstrukce $\backslash rlap{\backslash kern\backslash x \backslash raise\backslash y \backslash box\backslash labelbox}$ zařídí usazení značky do žádané pozice.

To vlastně ale není ještě celá pravda. Existuje jedenáct referenčních bodů spojených s každým boxem, jak ukazuje následující diagram:



Není těžké napsat makra, která umístí zvolený referenční bod boxu do bodu, který je se značkou spojen. Tedy například nápis „top right“ byl vysazen pomocí příkazu

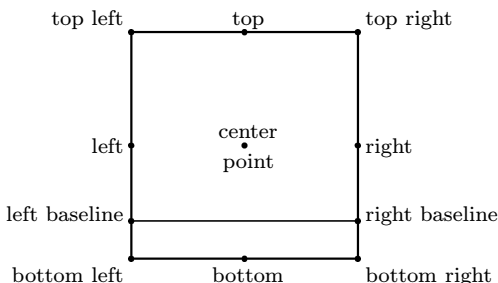
```
 $\backslash blpoint\{top\ right\}$ 
```

tj. levý dolní roh hboxu obsahujícího „top right“ má být umístěn ve značeném bodu.

A ještě to není celá historie. Čtenář, který prozkoumá obrázek podrobně, zaznamená, že značky se zdají být trochu natěsnané na značený bod. Je asi třeba ponechat navíc trochu místa. Příkazy typu $\backslash hskip2pt$ nebo $\backslash hskip-3pt$ posunou značku doprava nebo doleva, a je snadné definovat makra $\backslash up$ a $\backslash down$ taková, že (například)

```
 $\backslash up3pt$ 
```

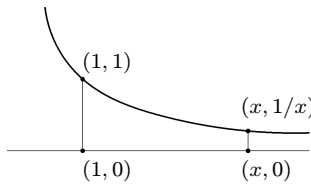
pomůže doladit vertikální umístění. Můžeme těchto úvah využít k vylepšení diagramu s referenčními body:



Následující tabulka vypisuje všechna zaměřovací makra a s nimi spojené orientace. Některé referenční body mají dvě makra. Uživatel, který potřebuje umístit levý horní roh značkového boxu, může použít $\backslash tlpoint$ i $\backslash ltpoint$, takže nevznikají problémy se zapamatováváním správného pořadí písmen.

Orientace	Jméno makra
vlevo nahoře	<code>\tlpoint</code>
	<code>\ltpoint</code>
nahoře	<code>\tpoint</code>
vpravo nahoře	<code>\trpoint</code>
	<code>\rtpoint</code>
vpravo	<code>\rpoint</code>
vpravo dole	<code>\brpoint</code>
	<code>\rbpoint</code>
dole	<code>\bpoint</code>
vlevo dole	<code>\blpoint</code>
	<code>\lbpoint</code>
vlevo	<code>\lpoint</code>
základní čára vlevo	<code>\point</code>
	<code>\lBpoint</code>
	<code>\lBpoint</code>
základní čára vpravo	<code>\rBpoint</code>
	<code>\rBpoint</code>
středový bod	<code>\cpoint</code>

S těmito pomůckami můžeme předělat Knuthův obrázek z *ACP*, tentokrát i se značkami:



Potřeboval jsem k tomu tato makra:

```

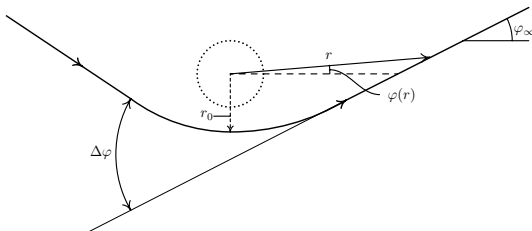
\font\figfont=1onx
\beginfig
\lpoint{$(1,1)$}%
\lpoint{\up2pt $(x,1/x)$}%
\tpoint{\down 2pt $(1,0)$}%
\tpoint{\down 2pt$(x,0)$}%
\endfig

```

Příklad

Coby závěrečný příklad následuje ještě obrázek vybraný z textu o obecné relativitě. Skutečný kontext či smysl obrázku není důležitý (vztahuje se k ohybu světla

způsobenému relativistickými účinky), dobře je však vidět řada zajímavých efektů, které METAFONT umožňuje.



Příklad ukazuje, že METAFONT umí nakreslit mnoho důležitých pomocných objektů, jako jsou šipky s libovolnou orientací, tečkované a čárkované čáry, atd.

Do zbraně!

Jedním z cílů tohoto článku je zažehnout oheň v srdcích horlivých mistrů METAFONTu. Komunita uživatelů $\text{T}_{\text{E}}\text{X}$ a METAFONTu by mohla mít užitek z vytvoření balíku maker, jenž by se měl k METAFONTu stejně, jako se má $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ nebo $\mathcal{A}\mathcal{M}\mathcal{S}\text{-T}_{\text{E}}\text{X}$ k $\text{T}_{\text{E}}\text{X}$. Kdo se hlásí?

Literatura

- [1] Hobby, John D. A METAFONT-like system with PostScript output. *TUGboat* 10(4), str. 505–512, 1989.
- [2] Simpson, Richard O. Nontraditional uses of METAFONT. Sborník *T_EX: Applications, Uses, Methods* (editor Malcolm Clark), Ellis Horwood Ltd., London, 1990, str. 259–272.
- [3] Wilcox, Patricia. Metaplot: Machine-independent line graphics in $\text{T}_{\text{E}}\text{X}$, *TUGboat* 10(2), str. 179–187, 1989.

(Překlad: Danuše Lhotková)

Popis formátu `dismac`

Tento příspěvek popisuje formát `dismac`, který se hodí pro přípravu rozsáhlejších matematických a technických textů, jako jsou diplomové a disertační práce, různé zprávy a s určitými drobnými úpravami také knihy. Tak, jak je zde popsán, je určen pro psaní českých textů, ale velmi snadno jej lze modifikovat pro psaní ve slovenštině, v angličtině nebo jiném jazyku.

Před tím, než jsem na tomto formátu začal pracovat, jsem pochopitelně zvažoval, zda se pro daný účel nedá použít některý z obecně dostupných formátů.

Formát `plain` je na příliš nízké úrovni a nenabízí prakticky žádnou podporu pro logické strukturování textu, křížové reference apod. Jeho smyslem totiž je poskytnutí základních nástrojů, s jejichž pomocí je možno efektivně budovat specializované