Program a2ac – manipulace s fontem na úrovni PostScriptu.

Petr Olšák

Abstract. V tomto článku je popsán autorův program `a2ac`. Program umožňuje sázet texty PostScriptovým fontem v jazycích, ve kterých se používají akcentované znaky. Výchozí fonty nemusejí obsahovat celou abecedu daného jazyka; stačí, když jsou přítomny jednotlivé akcenty samostatně. Konfigurační soubory programu `a2ac` jsou nezávislé na kódování PostScriptového fontu i na kódování, které je použito v sázecím systému. Program může být použit na přípravu fontů pro libovolný sázecí systém, ovšem testován byl pouze TEX. Použití `a2ac` pro TEX vytváří alternativu k programu `fontinst` od Alana Jeffreyho. Balík `a2ac` je volně k dispozici na CTANu včetně zdrojového textu v jazyce C. Program byl autorem testován na systémech SUN OS, Linux a dále na DOSu.

Program `a2ac` (Afm To Afm add Composites) konvertuje `afm` soubory (Adobe Font Metrics) na nové soubory rovněž ve formátu `afm`. Při své práci načítá tzv. definiční soubor, v němž je stanoveno, jaké změny se mají v `afm` formátu provést. Především se přidávají nové kompozitní znaky a rozšiřuje se tabulka kernů o nové kerningové údaje podle masek.

V metrice `afm` jsou informace o znacích vyjádřeny pomocí symbolických názvů znaků (např. `Scaron` znamená Š, `Uring` je Ů a `eacute` je é). Každý použitý název může být podchycen v metrice dvěma způsoby. Buď je vázán na konkrétní pozici ve fontu (kód a PostScriptová procedura na vykreslení znaku), nebo má kód nastaven na -1 a je popsán v metrice jako složenina jiných znaků. Popis složených (kompozitních) znaků se přitom opírá o symbolické názvy, takže není závislý na kódování fontu.

Základní myšlenka programu `a2ac` je založena na tom, že v definičním souboru popíšeme všechny potřebné složeniny pomocí symbolických znaků a program `a2ac` je do metriky doplní. Protože program pracuje na úrovni symbolických názvů znaků, jeho činnost je naprosto nezávislá na kódování jak PostScriptového fontu, tak vnitřního kódování sázecího systému, pro který se font připravuje.

Program `a2ac` je při čtení definičního souboru vybaven nenulovou inteligencí. Umožňuje deklarovat a používat proměnné, psát metrické a kompozitní informace pomocí jednoduchých výrazů a pomocí masky kernů doplňovat rozsáhlé množství nových kerningových informací pro nové znaky.

Následuje ukázka formátu definičního souboru. Příklad zahrnuje informace o nových písmenech É, Ě, é a ě. Nejprve jsou definovány znaky `Ecaron` a `Eacute` jako kompozitní (viz řádky NC), pak jsou vytvořeny pro tyto znaky nové kerningové informace. Kerny budou stejné, jako kerny s písmenem E (viz řádek NK). Dále jsou definovány znaky `ecaron` a `eacute` jako kompozitní a tyto znaky ponesou stejné kerningové informace, jako by se jednalo o písmeno e (viz řádky NC a NK). Nakonec jsou uvedeny některé výjimky. Dvojice Pě, Tě a Vě nebudeme kernovat, zatímco dvojice Pe, Te a Ve většinou mají záporný kern. Podobně pro dvojice Pé a Té.

```
NC Ecaron 2 ; PCC E 0 0 ; PAT caron 0 Carontop ;
NC Eacute 2 ; PCC E 0 0 ; PAT acute Acuteshift Acutetop ;
NK (Ecaron,Eacute) : E
NC ecaron 2 ; PCC e 0 0 ; PAC caron 0 0 ;
NC eacute 2 ; PCC e 0 0 ; PAC acute acuteshift vshift ;
NK (ecaron,eacute) : e
RK (P,T,V) ecaron 0
RK (P,T) eacute 0
```

Údaje o umístění akcentů jsou zde uvedeny pomocí proměnných (např. `acuteshift`), které jsou v definičním souboru deklarovány dříve. Program usazuje akcenty podle různých hledisek (viz `PCC`, `PAT` nebo `PAC`) navíc s přihlédnutím ke sklonu písma. Podrobnější informace o formátu definičního souboru najde čtenář v dokumentaci `a2ac-cz.doc`.

Do definičního souboru nelze zanášet informace o ligaturách. Ukážeme, že to nevadí. Pokud font ligatury má, pak ve vstupním `afm` souboru se na konci některých řádků s prefixem C vyskytuje údaj o ligatuře. Například:

```
C 102 ; WX 333 ; N f ; B 20 0 383 683 ; L i fi ; L l fl ;
```

Tato informace se přepíše do výstupního `afm` souboru a pro vytvoření záznamu v tabulce ligatur pro sázecí systémy je dostačující. V případě TEXu čte tyto údaje program `afm2tfm` a vytváří odpovídající záznam v ligační tabulce pro formát `tfm`. Navíc se dají prostřednictvím souborů typu `enc` definovat nové, speciálně TEXovské, ligatury. Například řádek v souboru `xl2.enc`

```
% LIGKERN hyphen hyphen =: endash ; endash hyphen =: emdash ;
```

definuje známé ligatury pro „`--`“ a „`---`“.

Příprava fontu pro TEX.

Po transformaci `afm` metriky programem `a2ac` se může použít program `afm2tfm`, který čte upravený soubor `afm`. Kromě toho program čte soubor `enc`, podle něhož provádí překódování fontu do kódování, které je použito v TEXu. Výsledkem je virtuální font, který obsahuje nejen informace o překódování fontu podle souboru `enc`, ale obsahuje též mechanismy na sestavení všech kompozitních znaků. Tyto mechanismy byly přitom definovány pomocí popisu kompozitních znaků v souboru `afm`. O to se postaral program `a2ac` prostřednictvím svého definičního souboru.

Začněme nyní s přípravou fontu pro TEX, který podporuje českou a slovenskou sazbu. Nejprve spustíme program `a2ac`. Ten se volá se třemi parametry:

```
a2ac input.afm corr.tab output.afm
```

První parametr je název vstupního (`afm`) souboru, druhý parametr označuje definiční soubor obsahující definice změn, které se mají provést. Třetí parametr je název výstupního souboru. Přípony souborů (`.afm`, `.tab`) je nutno uvést, program si je nedomyslí.

Příkazový řádek volání programu bývá obvykle součástí nějakého scriptu nebo dávkového souboru. Například, pro přípravu fontu z `afm` formátu do `tfm` můžeme použít jednoduchý UNIXový script, který vypadá takto:

```
a2ac $1.afm cscorr.tab c$2.afm
afm2tfm c$2.afm -t xl2.enc -v c$2 r$2
vptovf c$2.vpl c$2.vf c$2.tfm
```

Pokud se script jmenuje třeba `mkfnt`, pak nový font přípravíme například tímto povelem:

```
mkfnt Times-Roman ptmr
```

K balíku programu je přidán definiční soubor `cscorr.tab`, který obsahuje definice korekcí, pomocí nichž lze vytvořit font obsahující znaky české a slovenské abecedy a vhodný kerning. Při použití tohoto souboru se pouze vyžaduje, aby vstupní `afm` soubor obsahoval všechny akcenty pro české a slovenské znaky. To splňují např. všechny standardní Adobe fonty. Výstupem je `afm`, obsahující navíc kompozitní znaky a rozšířený kerning.

Soubor byl beze změny použit na všechny základní PostScriptové fonty (32 fontů v každém RIPu) s velmi slušným výsledkem. Chceme-li docílit ještě lepšího výsledku, je možné pro různé fonty soubor `cscorr.tab` nepatrně obměňovat.

Po použití programu `a2ac` jsou další kroky pro přípravu fontu standardní. Například výše uvedený UNIXový script připravuje TEXovský font postavený na kódování $\mathcal{CS}$-fontů. Výstupen scriptu je `c*.tfm` (vstup pro TEX), a dále `r*.tfm` a `c*.vf` (vstup pro `dvips`). Jediné, co je třeba udělat, je umístit tyto soubory do odpovídajících adresářů, nebo upravit script tak, aby to dělal za nás. Dále je potřeba přidat jeden řádek do souboru `psfonts.map` (konfigurační soubor programu `dvips`). Řádek vypadá takto:

```
rfont PostScript-Font
```

Pokud není PostScriptový font instalován v RIPu výstupního zařízení, musíme jej uložit ve formátu `pfb` (`pfa`) do našeho počítače. Pak řádek v souboru `psfont.map` bude mít tvar:

```
rfont PostScript-Font </cesta/k/souboru.pfb
```

nebo (pro DOS):

```
rfont PostScript-Font <d:\cesta\k\souboru.pfb
```

Nyní můžeme zavést font do TeXu pomocí těchto příkazů:

```
\font\newfont=cfont
{\newfont zde je použit osmibitový text
          ve stejném kódování, jako je definováno v souboru
          xl2.enc. Tento text bude vysázen v novém fontu.}
```

Po TeXování a zpracování programem `dvips` dostaneme text vysázený novým fontem.

Kódování TeXu.

Není nutné, aby se kódování vstupního souboru shodovalo s kódováním TeXu. Je možné provést transformace na úrovni `tcp` tabulek v emTeXu nebo změnami vektorů `xord`/`xchr` v souboru `tex.ch` (změny zdrojového textu TeXu ve formátu `web`) nebo nastavením aktivních znaků. Soubor `enc` určuje kódování virtuálního fontu, které se musí shodovat s kódováním tabulek dělení slov. Tomuto kódování říkáme vnitřní kódování TeXu.

V našich zemích se asi nejčastěji používá vnitřní kódování TeXu podle CS-fontů (tzv. CS-kódování). Tyto fonty jsou přirozenou součástí pravděpodobně nejrozšířenější instalace TeXu u nás s názvem CSTeX. CS-fonty v tomto balíčku zcela nahrazují textové Computer Modern fonty. CS-kódování je nadmnožinou kódování textových fontů Computer Modern, přičemž rozšíření zahrnuje akcentované znaky české a slovenské abecedy podle ISO-8859-2.

Výhody tohoto kódování (ve srovnání například s mezinárodně používaným kódováním T1) jsou tyto:

- Fonty obsahují jen ty znaky, které jsou nutně potřeba pro českou a slovenskou sazbu. Příprava sazby tedy není zatížena zbytečnostmi, což četří čas i místo na disku. Například při generování z METAFONTových zdrojů se pracuje jen se znaky, které se vyskytují v české nebo slovenské sazbě a soubory `pk` jsou menší. Výjimky (třeba znaky v cizích jménech) lze vždy obejít použitím makra z plainTeXu nebo primitivem `\accent`.
- Protože je toto kódování nadmnožinou kódování textových fontů Computer Modern, je možno v instalaci zcela nahradit Computer Modern fonty CS-fonty bez dalších zásahů do TeXovských maker. TeX se pouze prostřednictvím `\catcode` seznámí s novými písmeny (kódy větší než 128) a pomocí `\lccode`, `\uccode` se naučí dvojice malé/velké písmeno. V instalaci CSTeX je například formát `csplain`, který je zcela shodný s formátem plain s jedinou výjimkou. Textové fonty CM jsou nahrazeny CS a formát je přizpůsoben pro osmibitový vstup v kódování ISO-8859-2.
- Protože jsou akcentované znaky umístěny podle ISO-8859-2, je možno v UNIXových instalacích rovnou použít osmibitový český a slovenský text bez vstupního překódování. V jednoduchosti je síla! Na druhé straně v DOSu není pro naše jazyky ustálen jediný standard kódování textových souborů. Proto je zde potřeba použít tabulku `tcp`.
- Kerning a další náležitosti fontu můžeme připravit s ohledem na českou a slovenskou sazbu a nemusíme čekat, až nám to někdo daleko za našimi hranicemi udělá. Principiálně ani není možné, aby jediný font vyhovoval tradicím většího množství národů současně.

V balíku programu `a2ac` jsou zařazeny soubory `xl2.enc` a `xt2.enc`. Tyto soubory definují kódování, které je nadmnožinou CS-kódování. Rozšíření zahrnuje některé znaky, které nejsou obsaženy v české a slovenské abecedě (tj. v CS-fontu), nicméně jsou zahrnuty v materiálu PostScriptových fontů kódovaných podle Adobe StandardEncoding.

Protože textové Computer Modern fonty nejsou jednoznačně kódované, je třeba řešit několik drobných problémů. Pro některé pozice existují dvě alternativy: rozlišují se fonty typu `\rm` a `\tt` (např. ligatura fi, nebo šipka dolů) a fonty typu `\rm` a `\it` (dolar nebo libra na stejné pozici).

Je zřejmé, že konzervativní rozšíření CM fontů (například CS-fonty) také nemůže být jednoznačné. Máme-li vnitřní kódování TeXu postavené na bázi CS-fontů, je třeba použít dva kódovací soubory typu `enc`: soubor `xl2.enc` pro fonty typu `\rm`, `\it` a soubor `xt2.enc` pro fonty typu `\tt`. Dolar je ponechán na pozici 36 ve všech fontech a libra je umístěna na pozici 132. Po zavedení PostScriptového fontu typu `\it` je tedy potřeba vědět, že libra není dosažitelná standardním plainTeXovským špinavým trikem: `{\it\$}`, ale je potřeba použít makro, které volá `\char132`.

Pozice 32 není v souboru `xl2.enc` definována, protože škrtátko pro polské L a l není v Adobe StandardEncoding zahrnuto. Místo toho jsou na pozicích 163 a 179 samostatné znaky L a l. Pokud tyto znaky chceme používat v plainTeXu s PostScriptovým fontem, je vhodné předefinovat makra `\l` a `\L`.

Všimneme si dále, že v Adobe StandardEncoding tabulce chybí trojnásobné ligatury a velká řecká písmena. Nepřítomnost trojnásobných ligatur v PostScriptovém fontu nám příliš nevadí, protože tyto ligatury se v českém a slovenském jazyce nevyskytují. Pokud navíc máme PostScriptový font v kódování ExpertEncoding, můžeme trojnásobné ligatury přidat na úrovni `vpl` souborů.

Nepřítomnost řeckých znaků je jedna z příčin (bohužel ne jediná), proč nelze snadno použít nový PostScriptový font v matematické sazbě. Velké množství plainTeXovských maker a vestavěných algoritmů TeXu se opírá o jisté speciální vlastnosti matematických fontů. Mám na mysli třeba rozsáhlé množství parametrů `\fontdimen` pro matematický font nebo různé plainTeXovké triky skládání dlouhých šipek pomocí segmentů složených ze znaku „mínus" nebo „rovná se". Aby vše vyšlo i pro PostScriptový font, je potřeba nejen zařadit na úrovni `vpl` do fontu písmena řecké abecedy (obvykle z fontu Symbol), ale je potřeba též obohatit font o plno parametrů `\fontdimen` a vše sladit s TeXovskými makry. To bohužel není jednoduchá záležitost a už vůbec se to nedá automatizovat. Odladit matematickou sazbu v novém fontu vyžaduje cit výtvarníka a hluboké zkušenosti s TeXem. Proto se často od sazby matematiky v PostScriptovém fontu upouští* a nové fonty se zavádějí jen do textové části sazby. Pokud pro zavedení fontu použijeme pouze primitiv `\font` a nebudeme se pouštět do zavádění nových matematických skupin pomocí `\textfont`, `\scriptfont` a `\scriptscriptfont`, zůstane při použití plainTeXu matematika ve fontu Computer Modern.

Náhradní fonty.

Pro prohlížeč a tiskárny nevybavené PostScriptem je většinou potřeba vytvořit náhradní font. Buď se spokojíme s přibližnou náhradou pomocí standardních CM (CS) fontů nebo použijeme program `ps2pk`. Druhá cesta je možná jen za předpokladu, že font máme k dispozici ve formátu `pfb` nebo `pfa` v počítači. To nemusí být pravidlem – často je font instalován jen v RIPu výstupního zařízení a nejsme schopni jej odtud „vydolovat". Proto popíšeme cestu, jak je možné pracovat s náhradními fonty.

Náhrada běžnými CS-fonty se provede pomocí následujících kroků. Například pro nahrazení fontu `cptmr` použijeme font `csr10`.

```
tftopl csr10.tfm csr10.pl
```

provede konverzi metriky do formátu `pl`.

---

* Existují PostScriptové fonty, které jsou vybaveny podporou sazby TeXovské matematiky, ale je jich málo.

Dále je třeba do `csr10.pl` doplnit příkazy MAPFONT a VTITLE (viz [1]). Na začátek souboru napíšeme

```
(VTITLE Substitute font cptmr -> csr10)
(MAPFONT D 0
  (FONTNAME csr10)
  (FONTCHECKSUM D 0)
  (FONTAT R 1.0)
  )
```

Nyní přejmenujeme pozměněný spoubor `pl` na `vpl`:

```
mv csr10.pl ptrm.vpl
```

(v DOSu: `ren`), a konečně vytvoříme „nahrazovací“ virtuální popis:

```
vptovf ptmr.vpl ptmr.vf ptmr.tfm
```

Soubor `tfm` zahodíme a soubor `vf` předložíme ovladači, který má použít náhradní font. Pozor! Nikoli pro finální výstup, tedy nikoli pro ovladač `dvips`.

O názvech akcentovaných znaků.

Pro znaky ď, ť by se správně měly použít symbolické názvy `dquoteright` a `tquoteright`. Místo nich byly ale použity symbolické názvy `dcaron` a `tcaron`. Důvod je ten, že tyto znaky mají svůj protějšek ve verzálkách nazvaný správně `Dcaron` a `Tcaron`. Kdybychom nazvali písmeno ď jako `dquoteright`, pak by použití parametru `-V` v programu `afm2tfm` pro vytvoření fontu „Capitals and Small Capitals“ nedalo správný výsledek. Abychom nezaváděli nová symbolická označení pro Ľ a ľ, zůstali jsme i zde u názvu `Lcaron` a `lcaron`.

Reference

[1] Donald Knuth: Virtual fonts, a more fun for grand wizards. TUGboat 11(1):13–23, April 1990.
[2] Petr Olšák: Typografický sytém TEX. CSTUG 1995, 270 pages. ISBN 80-901950-0-8.
[3] Petr Olšák: Úvaha o fontech v CSTEXu. TEXbulletin 3/93 (121–131).