

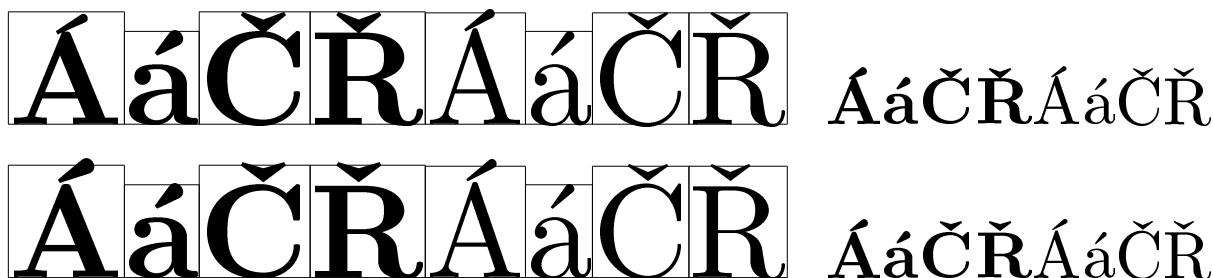
Návrh změn v \mathcal{C} fontech

Po diskusi na listu `ctex@` týkající se vztahu \mathcal{C} fontů a LM fontů vypluly na povrch některé chyby v \mathcal{C} fontech. Většina těchto chyb se týká Type1 varianty \mathcal{C} fontů. Nejzávažnější chyba této varianty asi je, že zde není vhodná licence, neboť Type1 varianta byla převzata z BaKoMa fontů, které dovolují jen nekomerční použití. Tím se ukázalo, že bude asi potřeba v nejbližší době Type1 variantu \mathcal{C} fontů zcela předělat.

Při tvorbě nové Type1 varianty se zřejmě bude vycházet ze vzoru, kterým jsou METAFONTové zdroje \mathcal{C} fontů. Zaměřil jsem se tedy nejprve na tyto zdroje a podrobil je revizi. Tento článek popisuje návrh změn \mathcal{C} fontů jen na úrovni METAFONTových zdrojů a zatím vůbec neřeší následnou Type1 reimplementaci. Předkládám tedy k diskusi zatím jen změny na úrovni METAFONTových zdrojů.

Na povrch vyplula i jedna závažná chyba v METAFONTových zdrojích \mathcal{C} fontů, kterou popsal pan Píška ve svém příspěvku pro list `ctex@` se subjektem „BUG in \mathcal{C} fonts: uppercase acute depends on resolution“. V reakci na tento příspěvek jsem podrobně popsal technické pozadí této chyby, takže zde se k tomu vrátím jen stručně. V METAFONTových zdrojích byl chybně uveden výpočet „tloušťky“ čárky nad verzálkami pro tučné varianty fontů tak, že tato tloušťka závisí na rozlišení, pro které se font generuje. Ve vyšších rozlišeních byla čárka tenká a nedomrlá, zatímco v nízkých rozlišeních byla v pořádku. Chybu vidíte na prvním řádku ukázky (níže) u prvního písmene **Á**. Ve větší verzi (vyšší rozlišení) je čárka hubená jak lunt zatímco na stejném řádku v menší verzi (poloviční rozlišení) je tatáž čárka dostatečně vypasená.

Tato chyba a celá diskuse kolem \mathcal{C} fontů mě vyprovokovaly znovu otevřít editorem METAFONTové zdroje a pokusit se udělat pár změn k lepšímu. Kromě uvedené chyby jsem se zaměřil ještě na korekce usazení akcentů vesměs nad verzálkami tak, aby to vypadalo lépe. Například jsem posunul verzálkové akcenty u serifových řezů mírně vzhůru, aby háčky nesplývaly s verzálkou ani u obloukových a tučných tahů (viz písmeno **Č**). To zdaleka nebyly všechny změny. V tomto článku se pokusím přesně popsat všechny navrhované změny.



V uvedeném obrázku je na prvním řádku stávající stav \mathcal{C} fontů a na druhém řádku vidíte navrhované změny. V této ukázce jsou fonty `csbx10` a `csr10`. Pro všechny ukázky v tomto textu jsem font zvětšil do velikosti 50pt s obrysy, které odpovídají metrikám. Dále jsem použil poloviční velikost 25pt bez obrysů. Fonty pro ukázky jsou generovány METAFONTem pro rozlišení tiskárny 600dpi, tj. pro tuto ukázku byly vygenerovány fonty `csbx10.3000pk`, `csbx10.1500pk`, `csr10.3000pk` a `csr10.1500pk` ve staré i nové verzi a takto byly zařazeny do tohoto dokumentu. Doprovodný text (mimo ukázky) je sázen starou Type1 obrysovou variantou \mathcal{C} fontů, která nyní není předmětem našeho zkoumání.

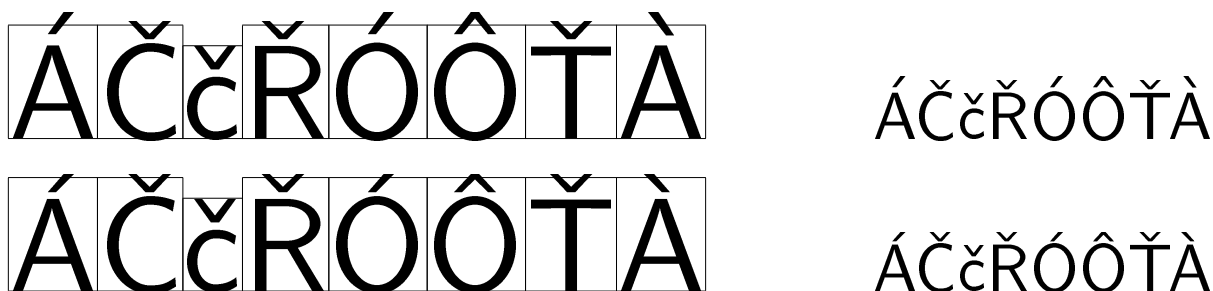
Metriky stávajících znaků jsem vůbec neměnil. Jde mi v prvé řadě o 100% zpětnou kompatibilitu výpočtu rozmístění znaků na stránku a už méně o vizuální zpětnou kompatibilitu, zvláště, pokud změny povedou k lepšímu. Vycházím z toho, že vizuální zpětná kompatibilita nemůže být nikdy zaručena, protože kdysi jsme dokumenty tiskli na starodávných jehličkových tiskárnách a dnes používáme vychytané laserovky. Musíme se smířit s tím, že ty dokumenty pak vypadají mírně jinak.

Nedovolil jsem si samozřejmě hnout ani s výškami znaků, ačkoli tyto rozměry se při výpočtu dokumentu projeví jen málokdy. Stačí ale, aby změna v tomto rozměru promluvila do výpočtu jen jedinkrát a celá kniha se může rozsypat na jiný počet stránek. To samozřejmě není přípustné. Vidíte tedy, že posunutí akcentů nad verzálkami nahoru jsem udělal za cenu toho, že akcenty budou mírně vyčnívat z rámečku daného metrikou. Ony už mírně vyčnívaly i v původní verzi, takže jsem se snad nedopustil moc velkého prohřešku. Dovedu si představit jedinou situaci, kdy na metrické výšce akcentového znaku záleží: pokud chceme nad tento znak pomoci primitivu `\accent` umístit druhý akcent. To snad není moc častá situace v našich zemích, takže nepatrné přetahy přes metriku snad nebudou tolik vadit.

Přiznám, že po otevření zmíněných METAFONTových zdrojů se mi udělalo poněkud smutno. Je to štipované jak ponožky. Původní Knuthovy zdroje jsou nejprve modifikovány poláky Bogusławem Jackowskim a Markem Ryćko. Na to asi navázali Petr Novák a další tvůrci ČSfontů. Každý přidal něco. Přitom se zdá, že některé vzorce nebyly vytvořeny při maximální duševní přítomnosti. Jsou tam dělány vážené průměry hrušek s jablky (lidově řečeno) a výsledek je použit jako pozice pro umístění akcentů. Aby to začalo mít logiku, musel bych to udělat zgruntu znovu. Na to ale nemám čas, takže jsem bohužel navázal na to štipování a přidal další záplaty. Ještě štěstí, že se do těch zdrojů moc lidí nedívá a fonty jsou hodnoceny jen podle výsledku.

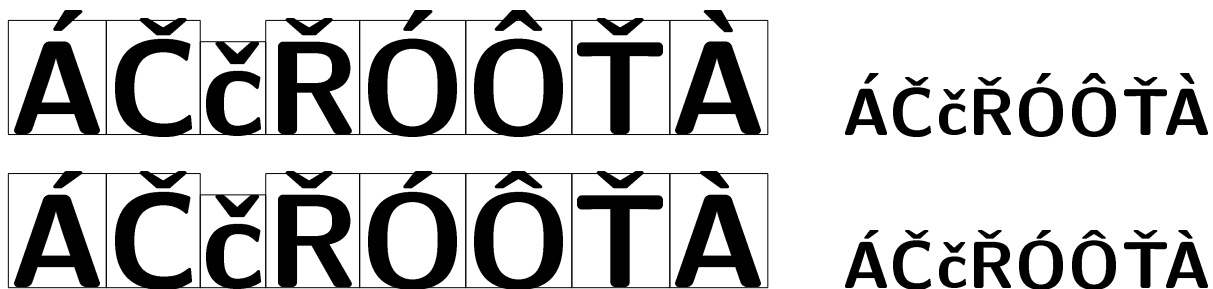
Problém je navíc v tom, že jeden program pro výpočet akcentu je použit ve všech 56 variantách jednotlivých řezů. Takže dloubat do toho znamená totéž, jako dloubat do živého organismu. Lokálně (v jednom řezu) se to spraví a ve čtyřiceti jiných se to stejným zásahem pokazí. Vyžadovalo to tedy při návrhu oprav pracovat obezřetně a s rozvahou. Také to vyžadovalo do úmuru provádět testy na všech 56 řezech současně a neustále je znovu a znovu všechny generovat. Tuto možnost Petr Novák v době vzniku ČSfontů kvůli tehdejšímu výkonu počítačů asi neměl.

Během chvíle experimentování na mě u bezserifové verze fontů vyštěkla další chyba:

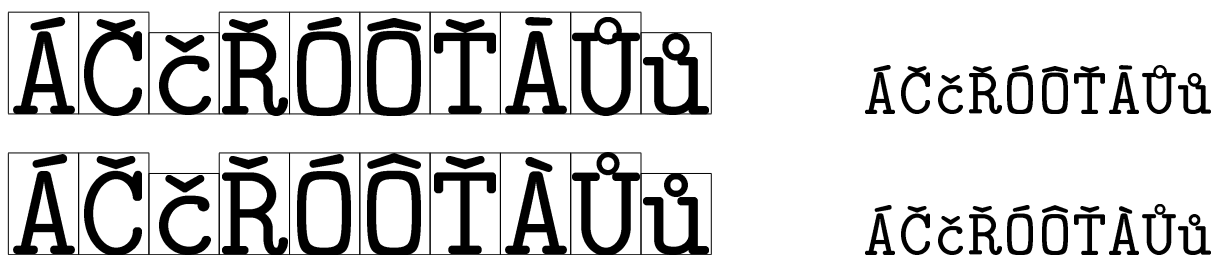


V prvním řádku vidíte původní font `csss10` a v druhém je návrh na opravu. Je zřejmé, že usazení čárek a stříšky horizontálně v původní verzi vůbec neodpovídá umístění háčku. Čárky vyčnívají až příliš přes rámeček daný metrikou a létají kdesi v prostoru. Dovolil jsem si tedy umravnit čárky a stříšku tak, aby zaujímaly nad verzálkou stejný prostor, jako původní háčky. Samotné háčky jsem neměnil.

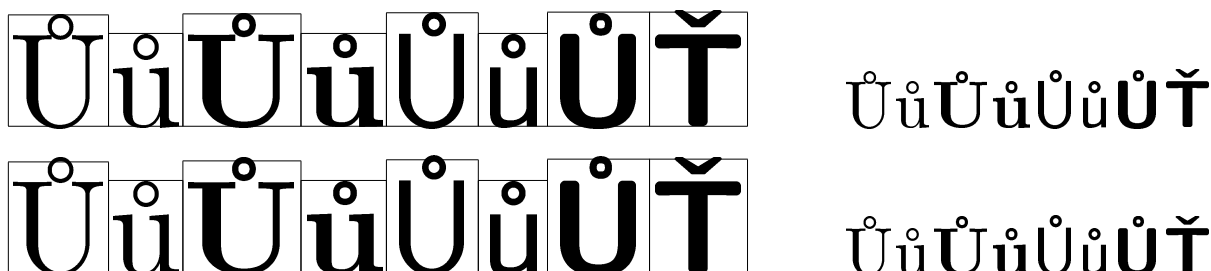
Vidíte, že stejná chyba je i v tučné verzi bezserifových řezů. Další ukázka obsahuje font `csssbx10`, první řádek původní font a druhý řádek návrh na opravu.



Strojopis (varianty `cstt*`) si také vyžádal menší korektury. V tomto případě navrhuji akcenty nad verzálkami mírně zvýšit. Například stříška nad O byla ve stavu, kdy docházelo ke slití barvy akcentu s písmenem. Dále navrhuji zmenšit kroužek nad U i u, protože původní kroužek mi připadá poněkud humpolácký a nevhodně zasahující do kresby písmene. Je nicméně pravda, že strojopis je celkově humpolácké písmo, takže ten původní kroužek se k němu hodil. To je věc názoru.



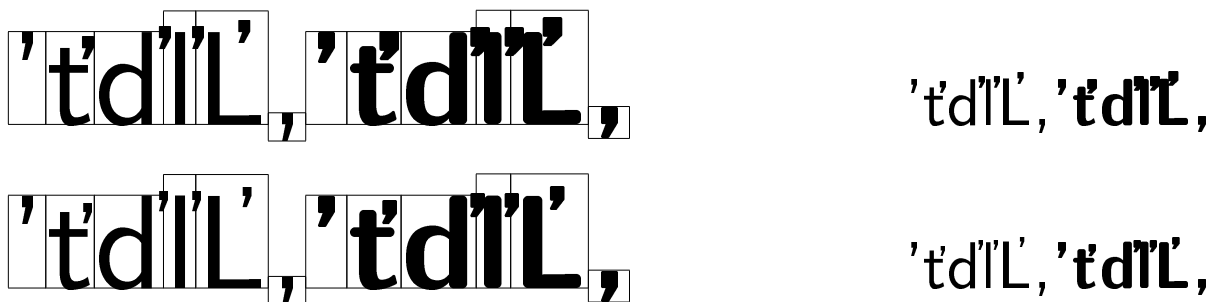
S kroužkem nad U jsem si pohrál i pro další řezy. Aby kroužek nezasahoval do kresby znaků, posunul jsem jej mírně nahoru. Stejně jsem kroužek posunul i u bezserifových verzí, aby jeho umístění mělo stejnou polohu, jako háček. Tvar kroužku jsem ale na rozdíl od fontu pro strojopis neměnil. Podívejte se na kroužek v řezech `csr10`, `csbx10`, `csss10` a `csss10`:



Akcenty pro písmena `ť`, `ď`, `l` a `L` jsou kapitolou samu pro sebe. Podle názoru pana Štorma je chybou, že jsou v mnohých fontech tyto akcenty odvozeny od apostrofu. Ve skutečnosti se jedná o zcela atypický akcent, který se apostrofu možná vzdáleně podobá. V původních `ČS`fontech sice nebyl tento akcent „doslova“ odvozen od apostrofu (měl kratší vlasový tah), ale zrno bylo stejně velké. To působilo potíže, zvláště pokud následovalo písmeno s dřikem po levé straně. Bohužel, tyto situace vůbec nejsou v `ČS`fontech řešeny dodatečným kernem (v tabulce kernových párů jsou jen znaky `ď`, `l`, `ť` následované pravou českou uvozovkou). Do tabulky kernů samozřejmě kvůli dodržení zpětné kompatibility zasáhnout nemohu. Rozhodl jsem se tedy významně zmenšit zrna uvedených akcentů. V ukázce je font `csr10` a `csbx10`. Prvním znakem v řadě písmen je apostrof (pro srovnání). Skutečnost, že `l` a `L` mají jinou výšku než `ť` a `ď` raději nebudu komentovat. Kvůli zásadě nesahat na metriky existujících písmen to ponecháme ve fontu jako „trvalou zvláštnost“. Výška znaku ovlivní sazbu jen málokdy a při záporném `\lineskiplimit` skoro vůbec ne.



Podobným způsobem jsem se rozhodl zmenšit akcent u `ť`, `ď`, `l` a `L` i pro bezserifové varianty. U tučné varianty to pořád působí potíže, ale je to určitě výrazně lepší, než v původní verzi.



Skloněné varianty jsou v `CM` fontech důsledně dělány pomocí
`currenttransform=identity slanted slant;`

takže většinou po úpravách a kontrole v přímých variantách nepřicházejí ve skloněných variantách žádné nové problémy. Překontroloval jsem tedy jen kurzívu, která má speciální kresbu. Liché řádky obsahují původní font a sudé řádky návrh změn.

Á á Č č Ř Ů ů

Á á Č č Ř Ů ů

Á á Č č Ř Ů ů

Á á Č č Ř Ů ů

À à Ô ô Ď ě Ľ

À à Ô ô Ď ě Ľ

À à Ô ô Ď ě Ľ

À à Ô ô Ď ě Ľ

V CM fontech najdeme mnoho dalších variant fontů, ve kterých Knuth projevil svou hravost, ale jinak nejsou moc k potřebě. Mám na mysli `cmdunh10`, `cmfi10`, `cmff10` a další. Čsfonty obsahují úplnou kopii textových CM fontů, takže obsahují i tyto varianty. Přiznávám se, že jsem si s těmi variantami příliš nehrál. Takže jsem moc nezkoumal, jak změny ovlivní polohu akcentů v těchto řezech. Byly to řezy extrémně zvláštní a budou i po změnách. Není podle mého názoru nutné kvůli těmto řežům zakládat do programů pro pozice a tvary akcentů nějaké výjimky. Například akcenty ve fontu `cmdunh10` létají dosti vysoko nad verzálkami, ale to ke zvláštnosti toho fontu asi patří.

Pro ilustraci zde uvedu jen font `cmfi10`, aby bylo vidět, jak se na něm projeví navrhované změny. Vidíte, že se projeví překvapivě. Ačkoli jsem globálně háčky nad verzálkami zvedal, zde ze záhadných důvodů klesly. To je důkaz, že METAFONTový zdroj Čsfontů je skutečně živý organismus. Nemíním se tím dále zabývat. Font je kuriózní, takže může mít i kuriózní háčky.

Ā ā Ŕ Ů ů ě Ľ

Ā ā Ŕ Ů ů ě Ľ

Ā ā Ŕ Ů ů ě Ľ

Ā ā Ŕ Ů ů ě Ľ

Další zvláštností je varianta `csinch` odvozená od fontu `cminch`. Font obsahuje jen verzálky a číslice ve velikosti 1 palec. Chybí mu cokoli dalšího, neobsahuje ani interpunkci. Domnívám se, že font je plně nahraditelný fontem `csssbx10 at 1in` s tím rozdílem, že v tomto fontu máme kromě verzálek i všechny ostatní znaky. Myslím si, že smysl existence fontu `cminch` zanikl vyšrotováním starých počítačů, ve kterých se šetřilo místem na disku, a proto tam bylo vhodné ušetřit na velikosti PK souborů pro titulkové písmo tím, že se použijí jen verzálky. Dnes tento argument není podstatný. Přesto Čsfonty (asi z historických důvodů) tuto variantu podporují. Ukázalo se, že ji podporují nedostatečně. Pokud má font mít všechny verzálky, pak v současné verzi chybí Ů a slovakům jistě bude chybět Ô a přehlasované verzálky. Považuji to za chybu fontu `csinch`, takže jsem se rozhodl ji napravit a font doplnit o tyto chybějící verzálky: Ů, Ô, Ā, Ē, Ä, Ö, Ů. Interpunkci a další znaky jsem nepřidával. Kdo to bude potřebovat, stejně použije místo `csinch` asi font `csssbx10 at 1in`

V rámci změn navrhuji přidat do Čsfontů dva nové znaky: paragraf (`/section`) a znak měny Evropské unie. V diskusi kolem Čsfontů zaznělo, že takové změny v kódování jsou důvodem k zavrnutí Čsfontů kompletně. Nemyslím si to. Přidání nových a potřebných znaků do prázdných míst vůbec neohroží zpětnou kompatibilitu. Kdo nechce, nadále tyto znaky nemusí používat. Jediný problém tuším v tom, že prázdná místa Čsfontů si jednotliví uživatelé soukromě zaplnili svými znaky a mohou mít s novou verzí Čsfontů z tohoto důvodu problémy. Tady byla chyba, že v rámci kódování Čsfontů nebylo rozhodnuto,

kteřé prázdne pozice jsou rezervovány pro budoucí použití a které jsou volné k soukromému zaplnění uživatelem.

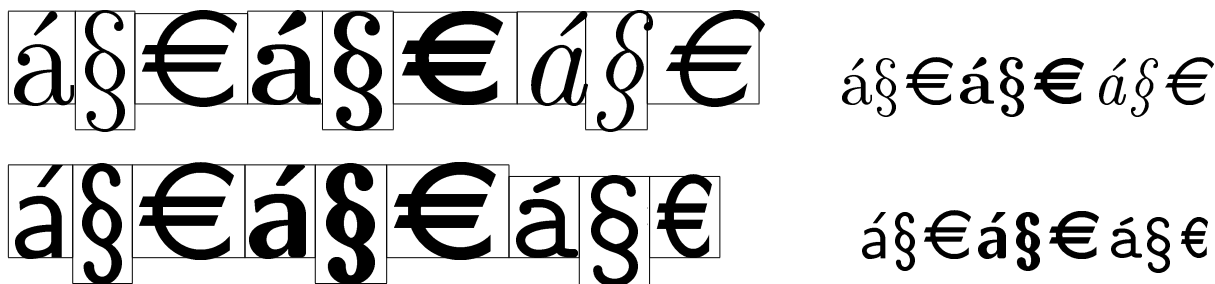
Kdo chce, může si napsat na začátek dokumentu „bezpečnostní kód“ například tohoto tvaru:

```
\setbox0=\hbox{\char134}
\ifdim \wd0=0pt \def\euro{Euro}
\else \chardef\S=167 \chardef\euro=134 \fi
```

Tím bude mít zaručeno, že mu dokument využívající sekvence `\euro` anebo `\S` bude fungovat jednak v distribucích, ve kterých je ještě stará verze `CSfontů`, a druhak v distribucích s novou verzí, kde se použijí nové znaky z fontu. Změnu ocení možná právníci, kteří používají znak paragraf většinou ve všech možných řezech (tučný, kurzíva atd.). Toto stará verze `CSfontů` neumožňovala, protože varianty tohoto znaku neexistovaly a makro `\S` se v plainu odkazovalo na matematický font s jedinou tvářív znaku paragraf. Nelze jej pak jednoduše měnit podle varianty aktuálního řezu. Znak Euro bude jistě také užitečné zahrnout do základního fontu. Vždyť možná zanedlouho budeme touto měnou běžně platit a dostávat v ní výplaty. Bylo by pak hloupé tisknout dokumenty s tímto znakem jen za pomoci externích záplatovacích makrobálíčků.

Otázkou pouze zůstává, na které prázdne pozice tyto dva nové znaky umístit. Navrhuji pro znak `/section` použít pozici 167, protože na této pozici je kódovaný v ISO-8859-2 a je mu toto místo vyhrazeno i ve standardním kódovacím souboru `x12.enc`. Bohužel, kódování znaku Euro není takto jednoznačné. Navrhuji použít pozici 134, protože tuto pozici jsem použil pro Štormovy fonty. Bude ale potřeba doplnit slovo `/euro` na tuto pozici i do kódovacího souboru `x12.enc`.

V následující ukázce jsou po řadě fonty `csr10`, `csbx10`, `csti10`, `csss10`, `csssbx10` a `cstt10`. Všechny v nově navrhované verzi (první i druhý řádek).



Pro znak paragrafu jsem použil METAFONTový program pro paragraf přímo z CM fontů (z matematického řezu) a shledal, že kromě přidání kurzívív korekce nepotřebuje žádných dalších úprav. Docela rozumě se mění jeho tvář podle varianty písmového řezu.

METAFONTový program pro znak Euro jsem převzal ze souboru `euro-ce/euromac.mf: v1.2 (2 June 1998) -- (c) Harold W. de Wijn`. Byl jsem nucen jej přizpůsobit pro `CSfonty` a navázat na jejich běžné parametry. Zajímavá finta tohoto programu spočívá v tom, že se nejprve nakreslí kolečko s vodorovnými čárkami a pak se METAFONTovou gumou vymaže pravá část kolečka ve tvaru jakési bílé jedničky. Tím vzniká požadovaný tvar. Není tedy jednoduché udělat pro varianty `cstt*` znak Euro s oblými konci. Netvrdím, že by to nešlo, ale zatím jsem nad tím nedumal.

V rámci změn `CSfontů` navrhuji z balíčku `CSfontů` vyřadit veškeré metriky, které nejsou základní a jsou odvozeny ze Sauterovy extrapolace. Jedná se o 52 metrik, které jsou všechny vyjmenovány na konci úvodní kapitoly v dokumentu `cs-fonts.doc`. Ponechám jen 56 základních metrik, které mají svůj protějšek v CM fontech. Vysvětlím důvody, které mě k tomu vedou:

- Ne na všech distribucích funguje Sauterova extrapolace.
- Sauterově extrapolované fonty nebudou mít svůj Typ1 protějšek, takže je obtížné je použít na distribucích, které pracují pouze s Typ1 verzemi fontů (což jsou dnes skoro všechny distribuce).
- Efekt Sauterovy extrapolace není v běžných velikostech tak významný a dostačujícím způsobem je nahradí lineární zvětšování.

Uvědomuji si, že lidé, kteří vsadili na metriky ze Sauterovy extrapolace, mohou namítat, že jejich další nepodporou nebudou fungovat jejich staré dokumenty. Mohou si ale v takovém případě Sauterovu extrapolaci do distribuce doinstalovat a zařídít, aby chybějící metriky vznikaly dynamicky podle potřeby. Pokud Sauterovu extrapolaci v distribuci instalovánu neměli, tak jim i dříve ty metriky samotné byly

k ničemu. Uživatelé těchto extrapolovaných metrik musejí bohužel počítat s tím, že budou mít problémy s návazností na Type1 varianty fontů, které nikdy tyto metriky nepodporovaly a asi ani podporovat nebudou.

Sami můžete vyzkoušet \mathcal{C} sfonty s navrhovanými změnami. Uvítám připomínky a návrhy na vylepšení, jejichž efekt bude úměrný vynaloženému úsilí. Změny najdete v souboru:

`ftp://math.feld.cvut.cz/pub/olsak/csfonts2006/csfonts.zip`

V archivu jsou všechny metriky a dále následující soubory:

```
cs-fonts.doc ... aktualizovaná dokumentace
csaccent.mf ... programy pro akcenty
csadded.mf ... přidané programy pro paragraf a Euro
cscode.mf ... přidány nové dva kódy pro paragraf a Euro
kmtitle.mf ... oprava fontu csinch
```

Pokud tyto soubory s příponou `mf` vyměníte za stávající soubory stejného jména ve své instalaci, vymažete všechny původní `*.tfm` \mathcal{C} fontů z instalace a nahradíte novými a vymažete všechny `*pk` z instalace, které obsahují \mathcal{C} sfonty, máte instalovanou novou (zatím provizorní a neoficiální) verzi \mathcal{C} fontů. Je dále samozřejmě nutné přegenerovat formáty `csplain` a `cslatex`, aby byly načteny nové metriky.

Soubory `*.mf` obsahují podrobný záznam o každé změně. Většinou je v komentářích napsáno něco jako `% corrected by P.O. 12.2005`.

Pozor! jedná se jen o testovací verzi, která zatím nemá podporu v Type1 variantě \mathcal{C} fontů. Je potřeba ji tedy testovat jen na úrovni METAFONTové varianty. Kdo neví, co to znamená, ať si raději svou $\text{T}_{\text{E}}\text{X}$ ovou instalaci neboří těmito experimenty.

Následující příloha obsahuje ukázky všech 56 \mathcal{C} fontů ve staré i nové verzi v přirozené velikosti. První řádek každého řezu je tištěný starou verzí \mathcal{C} fontu a druhý řádek verzí novou. V acroreaderu asi uvidíte rozmlžená písmenka (protože se jedná o METAFONTové bitmapy), takže pro důkladnější kontrolu to bude lepší vytisknout.

csr17	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr12	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr10	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr9	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr8	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr7	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr6	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csr5	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csbx12	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csbx10	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>
csbx9	<p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ</p> <p>ÁáÉéĚěČčŘřÝýŮůÛütdlL'ÓóÔôŽžÀ€§</p>

csinch

Á Č
Á Č Ů ü ô