Dvojité hranaté závorky v matematice

V matematické sazbě rozeznáváme pět druhů závorek: oblé ( ), hranaté [ ], složené (svorky) { }, dvojité ⟦ ⟧ a lomené ⟨ ⟩.

Pavel Pop: *Ruční sazba,* učebnice pro stud. obor Polygrafie, 1983.

Před nedávnem jsem potřeboval zařadit do matematických textů dvojité hranaté závorky. Kolega Miroslav Dont sehnal na síti font `bbold` v METAFONTových zdrojových textech. Font obsahuje malá i velká písmena psaná dvojitým tahem, dále také řecká písmena, některé matematické operátory a zmíněné závorky. Autorem fontu je Alan Jeffrey.

U „dodávky“ jsem nenašel makro pro zavedení těchto fontů do TEXu. Přitom font obsahuje tabulky umožňující sazbu těchto závorek v libovolné velikosti a dále obsahuje plno zajímavých symbolů a operátorů. Myslím si, že by mohlo čtenáře zajímat, jak se takový font zařazuje do matematiky, aby vše fungovalo jak v základní, tak v indexové a podindexové pozici a aby byly využity všechny možnosti fontu.

Vytvořil jsem soubor `bb10.tex`, který obsahuje zavedení uvedeného fontu do „desetibodového“ dokumentu. Na obsahu tohoto souboru si ilustrujeme problematiku zařazení matematického fontu.

```
% Macro loads the bbold fonts for mathematics in 10pt size

                               % Example for 12pt size:
\font\bbtext=bbold10           % \font\bbtext=bbold12
\font\bbscript=bbold7          % \font\bbscript=bbold9
\font\bbscriptscript=bbold5    % \font\bbscriptscript=bbold7
\font\bbex=cspex10             % \font\bbex=cspex10 scaled 1200

\newfam\bbfam
\newfam\bebfam

\textfont\bbfam=\bbtext
    \scriptfont\bbfam=\bbscript
    \scriptscriptfont\bbfam=\bbscriptscript
\textfont\bebfam=\bbex
    \scriptfont\bebfam=\bbex
    \scriptscriptfont\bebfam=\bbex
```

```
\def\sixt#1{\ifcase#1 0\or1\or2\or3\or4\or5\or6\or7\or8\or9\or
 A\or B\or C\or D\or E\or F\fi}

\def\[{\delimiter"4\sixt\bbfam5B\sixt\bebfam02 }
\def\]{\delimiter"5\sixt\bbfam5D\sixt\bebfam03 }

\mathchardef\squarcap="1\sixt\bebfam46
\mathchardef\circlevee="1\sixt\bebfam50
... atd...

\def\bbmathcodes{%
   \mathcode`\*="2\sixt\bbfam2A \mathcode`+="2\sixt\bbfam2B
   \mathcode`\(="4\sixt\bbfam28 \mathcode`\)="5\sixt\bbfam29
   ... atd...
   \mathchardef\alpha="0\sixt\bbfam0B
   \mathchardef\beta="0\sixt\bbfam0C
   ... atd...}

\def\bb{\fam\bbfam\bbmathcodes\bbtext}
```

Je vidět, že jsou vlastně použity čtyři fonty — `bbold10` pro základní velikost, `bbold7` pro indexovou velikost, `bbold5` pro podindexovou velikost a `cspex10` obsahující speciální symboly Vpravo od příkazů `\font` za procenty je uveden příklad pro zavedení fontů pro dvanáctibodový dokument. Analogicky lze zavést i fonty pro základní velikost jedenácti bodů. Chceme-li udělat univerzálnější makro pro všechny velikosti (po vzoru LaTeXu), lze v místě příkazů `\font` zahrnout větvení podle přání uživatele. Všechny potřebné velikosti fontů jsou v dodávce `bbold` obsaženy.

Makro zavádí dvě nové „rodiny“ (`\fam`). Rodina `\bbfam` obsahuje velká a malá písmena latinské abecedy i řecké (𝔸 𝔹 ℂ 𝕒 𝕓 𝕔 α β …), některé speciální znaky (# $ % & ⟨ ⟩ \ + * //) a další. Tato rodina shrnuje uvedené znaky ve třech velikostech (základní, indexová a podindexová poloha). V této rodině jsou také „výchozí“ velikosti závorek ⟦ ⟧.

Druhá rodina `\bebfam` zahrnuje jedinou velikost fontu speciálních symbolů. Obsahuje „zvětšené“ formy závorek ⟦ ⟧ a sedm operátorů typu „suma“ ve velikosti pro textový styl i „display“. Jedná se o tyto operátory: ⊓ ◻ ‖ ⦀ ▯ ⦶ ⦷.

Definice `\def\sixt#1` definuje příkaz `\sixt`, který vrací šestnáctkový zápis čísla příslušné rodiny. V dalších definicích se výsledek tohoto příkazu použije jako jedna cifra šestnáctkového zápisu kódu jednotlivých matematických objektů.

Následuje definice příkazů `\[` a `\]`. Tyto příkazy budou vracet příslušné dvojité hranaté závorky ve všech velikostech i libovolně „natahovatelné“. Kód příkazu `\delimiter` čtěte např. takto: čtverka – jedná se o matematický objekt typu „open“ (otevírací závorka), `\sixt\bbfam5B` – výchozí velikost se bere z fontu rodiny `\bbfam` z pozice 5B a konečně `\sixt\bebfam02` – zvětšená velikost se vezme z fontu rodiny `\bebfam` z pozice 02 (hexadecimálně). V posledně jmenovaném fontu je uložena informace tvaru „řetěz postupně se zvětšujících závorek“, přičemž tento řetěz je zakončen odkazem na tři znaky fontu: horní část, dolní část a střední část závorky. V případě požadavku na obří závorku si TeX veme horní a dolní část a střed vyplní dostatečným počtem opakování střední části. Takové závorky vidíte po stranách tohoto odstavce. Stejný princip znáte z fontu `cmex10`, kde jsou uloženy „natahovací“ verze všech běžně používaných závorek.

Nyní například

```
$$  \left\[ {A+B\over C + \left\[ {D+E\over F} \right\]} \right\],
    \quad \[1\]_{\[2\]_{\[3\]}}  $$
```

povede na

$$\left\llbracket \frac{A+B}{C+\left\llbracket \frac{D+E}{F} \right\rrbracket} \right\rrbracket, \quad \llbracket 1 \rrbracket_{\llbracket 2 \rrbracket_{\llbracket 3 \rrbracket}}$$

Upozorňuji uživatele LaTeXu, že `{x\over y}` znamená totéž, co `\frac{x}{y}` a že označení dvojitých závorek sekvencemi `\[` a `\]` jim může způsobit komplikace, protože tyto sekvence jsou v LaTeXu použity pro vstup a výstup do/z matematického módu (jakási náhrada za dolary). Pokud tyto sekvence nikdy v tomto smyslu neužíváte, můžete je s klidným svědomím používat pro dvojité závorky (re-definovat původní příkaz). Ke kolizím nedochází.

Pokud si budete chtít uvedenou definici opsat, je vhodné upozornit na důležitost mezery za kódem příkazu `\delimiter`. Kdyby tam nebyla, pak se například `$\[1+1\]$` expanduje na

```
$\delimiter"485B9021+1\delimiter"585D903$
```

což je určitě něco jiného, než

```
$\delimiter"485B902 1+1\delimiter"585D903 $
```

Dále v makru následují definice operátorů typu „suma“. Například příkaz

```
\mathchardef\squarcap="1\sixt\bebfam46
```

čtěte: sekvence `\squarcap` se definuje jako matematický objekt typu „suma" (jednička v kódu), jehož menší (textová) verze se vezme z fontu rodiny `\bebfam` z místa 46 (hexadecimálně). Ve fontu existuje odkaz z menší verze na větší, takže se o to nemusíme v TEXu starat a například

```
\squarcap_{n=1}^\infty = \circlevee_0^\infty
```

povede uvnitř textu na $\sqcap_{n=1}^{\infty} = \bigcirc\!\!\!\vee_0^{\infty}$, zatímco v „display" stylu dostaneme

$$\prod_{n=1}^{\infty} = \bigcirc\!\!\!\!\vee_0^{\infty}$$

Konečně příkaz `\def\bb` v makru definuje přepínač písma `\bb`, který pracuje podobně jako například známý přepínač `\bf`. Použití příkazu `\bb` v textovém režimu způsobí přepnutí fontu díky příkazu `\bbtext` (příkaz `\fam\bbfam` se neprojeví), zatímco v matematickém režimu naopak „zabere" příkaz `\fam\bbfam` a příkaz `\bbtext` nemá vliv. Takže například text z naseho noveho fontu byl vypsán sekvencí

```
například {\bb text z~naseho noveho fontu} byl ...
```

Původní font nemá háčky a čárky. Pozorný čtenář knížečky „The Pamphlet on TEX Fonts" si jistě všiml, že font `bbold10` je demonstrován pouze na anglické citaci.

Zatím jsem neměl potřebu ani chuť háčky do metafontových zdrojů dopsat. Jinak na tom byl Karel Horák, který tuto věc dodělal. Proto je možné, že v této ukázce už uvidíte jeho háček i čárku.

Mnohem užitečnější asi bude používat font v matematickém režimu. Proto přepínač `\bb` navíc spouští příkaz `\bbmathcodes`, v němž se (lokálně — v rámci skupiny) předefinují matematické objekty jako např. `+`, `\alpha` a podobně. Takže třeba

```
$$  {\bb \alpha (A + B) } = \alpha (A + B) $$
```

vede na

$$\mathbb{\alpha} \llparenthesis \mathbb{A} + \mathbb{B} \rrparenthesis = \alpha\,(A + B)$$

V definici příkazu `\bbmathcodes` například

```
\mathcode`\*="2\sixt\bbfam2A
```

lze číst takto: symbol `*` v matematickém módu bude znamenat binární operátor (dvojka v kódu) braný z fontu rodiny `\bbfam` z pozice 2A (hexadecimálně). Podobně se čtou příkazy `\mathchardef`, které definují kontrolní sekvence `\alpha`, `\beta` atd. Tyto definice nemám vypsané důsledně všechny ani ve vlastním souboru maker, protože se mi jeví výhodnější tam zařazovat nové matematické kontrolní sekvence až v případě potřeby. Vyžaduje to pracovat s tabulkou fontu `bbold10` a s rozmyslem.

Poznamenejme ještě, že příkaz `\mathcodes` v definici přepínače `\bb` způsobuje, že `\bb` není „ryzím“ přepínačem. Použití této sekvence způsobem:

```
obyč. text \bb jiny text \rm obyčejný text
```

způsobí totiž trvalé předefinování matematických kódů. Je proto nutné buď psát

```
obyč. text {\bb jiny text} obyčejný text
```

nebo zrušit příkaz `\bbmathcodes` z definice sekvence `\bb`.

Na závěr se přiznám, že jsem ztratil hodně úsilí při vypátrání jedné „záludnosti“. Všimněte si, že v definici `\bbmathcodes` není u příkazu `` \mathcode`+=... `` použit backslash před znakem plus, jak se sluší a patří (viz definice všech ostatních kódů). Totiž kontrolní sekvence `\+` je definována v Plainu pro účely tabbingu jako `\outer\def`. To způsobuje, že nelze sekvenci `\+` použít uvnitř těla definice v *žádném* smyslu. Tj. ani pro definici matematického kódu znaku `+` .

Uvedené makro `bb10.tex` není zdaleka univerzální. Na to přijdou velice rychle všichni uživatelé LaTeXu. Pokud například píšete poznámky pod čarou v petitu a v těchto poznámkách se vyskytne nějaký symbol z fontu `bbold`, je třeba použít font `bbold8`, přičemž ten není makrem `bb10.tex` zaveden. Podobně, pokud se speciální symbol vyskytne ve zvětšeném nadpise, je nutno zavést příslušně zvětšené fonty. Technika zavádění fontů automaticky podle požadované velikosti je vyřešena v LaTeXu (soubor `lfonts.tex`) a ještě lépe je řešena v NFFS. Kdo chce, může se v těchto makrech inspirovat. Sám si myslím, že použití speciálního fontu v nadpise či zmenšené poznámce je poměrně málo časté. Proto je možné takový výskyt řešit individuálně. Pokud ale navrhujeme formát knihy (kde např. všechna cvičení budou sázená menším písmem), je nutno celou věc do makra zahrnout a vše pořádně domyslet. Makrojazyk TeXu nám neklade žádná omezení.

Článek jsem dal číst kolegovi Dontovi, který nedávno přinesl zajímavou zprávu pro doplnění. Na síti nalezl balík, zvaný `St Mary's Road`, v němž je zahrnut font se zmíněnými závorkami (autorem je Alan Jeffrey společně s Jeremy Gibbonsem). Navíc tam najdete LaTeXovské styly pro zavedení fontu do dokumentu a font obsahuje podstatně větší množství nových matematických znaků. Stačí tedy pouze uvést příslušný název stylu a o příkazy typu `\mathcode` apod. se nemusíte starat. Chtěl jsem ale ve svém článku ukázat, že zavedení fontu na úrovni TeX-primitivů nemusí být zase tak komplikované, třebaže se jedná o matematický font. Výhodou je, že přesně víte, co máte zavedeno a jak.

17.1.1993 *Petr Olšák*