

David Nečas (Yeti)

# Yetiho typografický bestiář

Verze: 19. 5. 2003

Kvalitní matematická sazba je jednou silných stránek  $\text{T}_{\text{E}}\text{X}$ u. Bohužel to neznamená, že použití  $\text{T}_{\text{E}}\text{X}$ u automaticky vede k dokonalým výsledkům (jak se možná někdo mylně domnívá). Matematická symbolika je věc natolik složitá, různorodá a subtilní (a místy i obskurní), že automaticky dokonalou matematickou sazbu nelze očekávat od žádného počítačového programu; určitá míra lidské intervence je vždy nutná.

Cílem Bestiáře je popsat nejběžnější situace, kdy pro správné vysázení matematických výrazů nestačí napsat správně jména symbolů ve správném pořadí a zbytek nechat na  $\text{T}_{\text{E}}\text{X}$ u, a poskytnout praktické návody, jak  $\text{T}_{\text{E}}\text{X}$ u pomoci vysadit takovéto výrazy správně. Z množiny pravidel matematické sazby přijde tedy na přetřes pouze ta podmnožina, která může při použití  $\text{T}_{\text{E}}\text{X}$ u způsobovat potíže.

*Toto je alfa verze. Má k úplnosti, neřku-li dokonalosti, dost daleko. Pokud máte zájem na zlepšení tohoto stavu, pošlete bugreporty a nebo nejlépe kus sami napište – viz Licenci.*

# Obsah

<b>Makra používaná v bestiáři</b>	<b>3</b>
<b>1 Co se ve vzorcích sází antikvou</b>	<b>5</b>
1.1 Značky jednotek . . . . .	5
1.2 Značky funkcí . . . . .	5
1.3 Značky operátorů a operací . . . . .	6
1.4 Binární operátory . . . . .	7
1.5 Nematematické objekty . . . . .	8
1.6 Slova a části slov . . . . .	8
1.7 Indexy a exponenty . . . . .	9
1.8 Konstantní symboly . . . . .	10
<b>2 Mezery v matematice</b>	<b>12</b>
2.1 Automatické mezerování . . . . .	12
2.2 Operátory, funkce a jejich argumenty . . . . .	13
2.3 Interakce vzorců s větami . . . . .	16
2.4 Symboly, které se pletou . . . . .	17
2.5 Mezery z estetických důvodů . . . . .	20
<b>3 Sazba číselných hodnot</b>	<b>22</b>
3.1 Obecné zásady . . . . .	22
3.2 Úvodní příklad . . . . .	23
3.3 Písmo jednotek . . . . .	23
3.4 Mezery . . . . .	24
3.5 Anglosaské konvence . . . . .	26
3.6 Procenta, promile, stupně, grady a jiné speciality . . . . .	26
3.7 Peněžní hodnoty . . . . .	27
3.8 Časové údaje a kalendářní data . . . . .	27
<b>Licence</b>	<b>29</b>

## Makra používaná v bestiáři

Všechny příklady v Bestiáři by měly fungovat v Plain T<sub>E</sub>Xu i v ostatních běžných formátech. V ukázkových zdrojácích nicméně používám několik příkazů kolidujících se stejnojmennými příkazy L<sup>A</sup>T<sub>E</sub>Xu (jde ovšem o příkazy, jejichž užitečnost je diskutabilní, a není proto žádná zvláštní škoda je předefinovat) a několik nových příkazů. Obojí uvádím zde.

FIXME: některá uvedená makra se v textu vůbec nevyskytují (zatím?).

- \, Význam \, v matematice se nemění. V textu sází nezlomitelnou zúženou mezeru (`\narrowospace`) o velikosti 2/9 em s roztažností 1/6 em a stlačitelností 1/9 em. V L<sup>A</sup>T<sub>E</sub>Xu je v textu \, synonymem pro `\thinspace`, tj. pevnou mezeru o velikosti 1/6 em, kterou ovšem potřebujeme jen velmi řídké (jako na oddělení vnořených uvozovek ‘\, ‘).

```
\def\narrowospace{0.22222em plus0.16667em minus0.11111em }
\def\,{\ifmmode\mskip\thinmuskip\else\nobreak
\hskip\narrowospace\fi\relax}
```



Zúžená mezer by měla mít šířku mezislovní mezery minus polovinu šířky tečky, což zde není příliš dodrženo – v CM fontech je totiž tečka poměrně široká a mezislovní mezer zase úzká, takže by nejmenší šířka zúžené mezery vyšla příliš malá. Zvolenou hodnotu lze považovat za rozumný kompromis.

- \: V matematice je \: v L<sup>A</sup>T<sub>E</sub>Xu synonymem `\>` (mezer 4 mu s roztažností 2 mu a stlačitelností 4 mu). Zde se mění na mezeru o stejné velikosti 4 mu, ale s nulovou roztažností a stlačitelností; v textu \: sází ekvivalentní mezeru textovou.

```
\def\:{\ifmmode\mkern4mu\else\kern0.22222em\fi }
```



Zajímavou možností je použít v textu k tomuto účelu podržítka, které nastavíme jako aktivní znak, což nám umožní psát `12_450_Kč`, `60_km`, `16_777_216`, apod. (podržítka se tím ovšem stane křehkým).

```
\catcode'\_=\active
\def_{\relax\ifmmode\sb\else\kern0.22222em\fi }
```

- \( Příkaz \(\ sází levou kulatou závorku automatické velikosti. V L<sup>A</sup>T<sub>E</sub>Xu je použit pro začátek matematického prostředí.

```
\def\left{}
```

- \) Příkaz \) sází pravou kulatou závorku automatické velikosti. V L<sup>A</sup>T<sub>E</sub>Xu je použit pro konec matematického prostředí.

```
\def\right{}
```

`\d` Příkaz `\d` sází písmeno `d` antikvou.

```
\def\d{\rm d}
```

`\e` Příkaz `\e` sází písmeno `e` antikvou.

```
\def\d{\rm e}
```

`\.` Význam příkazu `\.` v textu se nemění (akcent `á`), ale v matematice `\.` sází tečku (na řádku) jako binární operátor (stejně jako pomocný příkaz `\dop`). Pokud `\.` jako akcent nepoužíváme, je mnohem lepší jednoduše jej předefinovat a ušetřit si tak trable s dalším křehkým (*fragile*) makrem.

```
\let\plaintexdot=.
```

```
\mathchardef\dop="213A
```

```
\def\.{\ifmmode\dop\else\expandafter\plaintexdot\fi}
```

Kdo naopak nepoužívá `\dop` a preferuje jako operátor násobení `\cdot`, necht' si představuje `\dop = \cdot`.

`\B` Penalta za povolený, ale ne právě vhodný zlom v úzké mezeře (např. v datu).

```
\def\B{\penalty1000 }
```

`\~` Makro `\~` sází českou pomlčku – tj. *en-dash* se zúženými mezerami okolo, přičemž jen druhá z nich je zlomitelná.

```
\def\~{\nobreak\hskip\narrowsspace\char'173
```

```
\hskip\narrowsspace}
```

`\=` Makro `\=` sází spojovník.

```
\def\={\discretionary{-}{-}{-}}
```

# 1 Co se ve vzorcích sází antikvou

V matematickém módu sází  $\text{T}_{\text{E}}\text{X}$  písmena  $a$  až  $z$  a  $A$  až  $Z$  implicitně matematickou italicou – tedy jako identifikátory  $a$  až  $z$  a  $A$  až  $Z$ . Prakticky všechno, co není identifikátor proměnné, se ovšem má sázet základním řezem písma – v matematických textech je jím téměř výhradně obyčejná antikva, což budu dále předpokládat.

Identifikátory proměnných přitom *nejsou*

- značky fyzikálních (a jiných) jednotek (m, kg, cm, V, A, dB, eV, kcal, GWh, MHz, kPa, torr, atd.)
- označení funkcí (abs, sin, cos, tg, arccosh, exp, log, erf, Ei, atd.)
- označení operátorů a operací (grad, div, lim, min, sup, Tr, Dom, ker, cl, ext, d, D, mod, atd.)
- značky různých nematematických objektů (chemických prvků, elementárních částic, atd.)
- slova a zkratky, která vůbec neoznačují objekty – vyskytují se zejména v samostatných vzorcích a v indexech (pro, pokud, nebo, a, jinak, min, max, kin, atd.)
- konstantní symboly (i, e, id, atd.)

V následujících odstavcích se budu jednotlivým kategoriím věnovat podrobněji; jedno obecné pravidlo však mohu uvést už teď:

*Každý vícepísmenný symbol, který nevznikl kompozicí jednopísmenných symbolů, se sází antikvou.*

## 1.1 Značky jednotek

Fyzikální (a jiné) jednotky zde nebudu podrobně rozebírat, jednak proto, že poznat, co je značka nějaké jednotky, by nemělo být těžké, a hlavně proto, že sazbě číselných hodnot veličin je věnována samostatná kapitola Sazba číselných hodnot.

## 1.2 Značky funkcí

S funkcemi a operátory je to trochu těžší než s jednotkami. Nicméně platí, že značky všech standardních funkcí se sází antikvou. Příkladem mohou být sin (funkce sinus), ln (funkce přirozený logaritmus), tgh (funkce hyperbolický tangens) nebo arccosec (funkce „inverzní“ k funkci kosekans). Na všechny uvedené funkce se ovšem vztahuje také obecné pravidlo.

Ale pozor: standardní funkce jsou také funkce sférické, Besselovy, Neumannovy, Hankelovy či Jacobiho, všechny možné funkce chyb (*error-*

-*functions*), dále Hermiteovy, Čebyševovy, Legendrovy a čertvíčí polynomy, Fresnelovy, Fermiho, eliptické a jiné integrály; označení řady z nich je přitom jednopísmenné. Všechna příslušná Ci, Si, Ei, cn, sn, dn, Y, J, N, H, T, L, U, P, E, F (většinou s nejrůznějšími indexy dole i nahoře) se tedy sází antikvou.

Takže zatímco  $P_l^m(x)$  ( $P^{\sim}_m_l(x)$ ) je *nějaká* funkce  $P_l^m$ , symbol  $P_l^m(x)$  ( $\{\rm P\}^{\sim}_m_l(x)$ ) označuje přidružený Legendrův polynom, tj. funkci splňující diferenciální rovnici

$$(1 - x^2) \frac{d^2}{dx^2} P_l^m(x) - 2x \frac{d}{dx} P_l^m(x) + \left( l(l+1) - \frac{m^2}{1-x^2} \right) P_l^m(x) = 0$$

(a jisté další podmínky) a žádnou jinou.

Pro ty nejběžnější funkce jsou v Plainu nadefinovaná makra `\arccos`, `\arcsin`, `\dots`, `\tan`, `\tanh`, která jsou popsána v každé příručce. Takže `\sin B` vysází  $\sin B$ , kdežto ze `\$sin B` bychom dostali součin proměnných  $s$ ,  $i$ ,  $n$  a  $B$ , tedy  $\sin B$ . Nicméně už třeba „česká“ označení funkcí tangens a kotangens (`tg` a `cotg`) mezi předdefinovanými symboly nenajdete, natož potom nějaké zběsilé speciální funkce. Tyto symboly je naštěstí snadné dodefinovat. Například příkaz pro symbol funkce `sinc` nadefinujeme

```
\def\sinc{\mathop{\rm sinc}\nolimits};
```

`\mathop` zde zajistí správné mezerování (jako matematický operátor – viz kapitolu Mezery v matematice), `\rm` správný řez písma (tj. antikvu) a `\nolimits` umístění případných exponentů a indexů vždy vedle a nikoli nad a pod značku funkce – viz také dále.



V  $\LaTeX$ u je systémovější napsat do definice raději `\operator@font` než `\rm`, protože tím specifikujeme logický atribut písma (písmo, kterým se sází jména funkcí a operátorů) namísto fyzického (antikva). Nicméně `\operator@font` je interní příkaz  $\LaTeX$ u (jeho jméno obsahuje `@`), musíme proto před jeho použitím změnit kategorii znaku `@` na *letter* příkazem `\makeatletter` a poté ji vrátit zase zpátky na *other* příkazem `\makratother`.

### 1.3 Značky operátorů a operací

Obecné pravidlo o vícepísmenných symbolech se vztahuje i na různé (unární) operátory jako `lim` (limita), `inf` (infimum), `max` (maximum), `Tr` (stopa matice), `det` (determinant), `ext` (vnějšek množiny), `ker` (jádro zobrazení), `grad` (gradient), `div` (divergence), `rot` (rotace), atd. Sázíme tedy

$$\operatorname{rot} \operatorname{rot} \mathbf{A} = \operatorname{grad} \operatorname{div} \mathbf{A} - \Delta \mathbf{A},$$

zdroják:

$$\operatorname{rot} \operatorname{rot} \vec{A} = \operatorname{grad} \operatorname{div} \vec{A} - \Delta \vec{A},$$

kde předpokládáme  $\vec{A}$  sázející vektory vhodným typem písma a  $\operatorname{rot}$ ,  $\operatorname{grad}$ ,  $\operatorname{div}$  definovány podle vzoru  $\operatorname{def} \operatorname{rot} \operatorname{mathop}{\rm rot}$ .

A opět pozor: antikvou sázíme také  $d$  (diferenciál) a  $D$  (jeho vícerozměrná obdoba). V integrálech tedy patří sázet  $f(x) dx$  ( $f(x) dx$ ) a nikoli  $f(x)dx$  ( $f(x) dx$ ), obdobně u diferenciálů.

Do této kategorie dále patří symboly  $\operatorname{Re}$  a  $\operatorname{Im}$  označující reálnou a imaginární část a symbol  $\operatorname{P}$  označující hlavní hodnotu. Pro ně se však častěji používá kaligrafické, lomené či nějak jinak „vyšinuté“ písmo (např. standardní příkazy  $\operatorname{Re}$  a  $\operatorname{Im}$  dávají  $\mathfrak{R}$  a  $\mathfrak{I}$ ).



Pokud bychom chtěli, aby se náš nově definovaný operátor choval jako  $\operatorname{lim}$ , tedy aby byly případné indexy umístěny v display módu nad a pod symbol, vynecháme ve výše uvedeném příkladu  $\operatorname{nolimits}$  (nebo ho nahradíme  $\operatorname{displaylimits}$  nastavujícím explicitně implicitní chování). Mají-li být indexy umístěny nad a pod symbol i v odstavci, nahradíme ho  $\operatorname{limits}$ . Například symbol  $\operatorname{LIM}$  s použitím  $\operatorname{LIM} f(x) = q$  nadefinujeme

```
\def\operatorname{LIM}{\operatorname{LIM}\operatorname{limits}}.
```

## 1.4 Binární operátory

Občas jsou potřeba nejen unární, ale i binární operátory. Pro ten nejběžnější, mod, označující zbytek po dělení je v Plainu nadefinováno makro  $\operatorname{bmod}$  s použitím

$$(a+b) \operatorname{bmod} n = a \operatorname{bmod} n + b \operatorname{bmod} n,$$

což dává

$$(a + b) \operatorname{mod} n = a \operatorname{mod} n + b \operatorname{mod} n.$$

Symbol „mod“ se používá ještě jedním způsobem (snad ještě častěji než tím prvním)

$$x \equiv y + 1 \pmod{m^2},$$

pro který je nadefinováno makro  $\operatorname{pmod}$ , takže zdroják předchozího příkladu je

$$x \operatorname{equiv} y + 1 \operatorname{pmod}{m^2}.$$

Symbol  $\operatorname{bdiv}$ , chovající se stejně jako  $\operatorname{bmod}$ , můžeme jednoduše nadefinovat

```
\def\operatorname{bdiv}{\operatorname{mathbin}{\operatorname{div}}}.
```



Podíváme-li se do zdrojáku Plainu, zjistíme, že skutečná definice `\bmod` je podstatně složitější – řeší vhodné chování symbolu v indexech a při lámání řádků, takže při vytváření vlastního podobného příkazu je lepší tuto definici zkopírovat a změnit název operátoru.

## 1.5 Nematematické objekty

Snad nejčastěji se vyskytujícími nematematickými objekty jsou značky chemických prvků. Sázíme tedy  $\text{NaCl}$  (chlorid sodný; sůl),  $\text{C}_7\text{H}_7\text{NO}_2$  (nitrotoluen) či  $\text{C}_2\text{H}_2$  (acetylen); nikoli  $\text{NaCl}$ ,  $\text{C}_7\text{H}_7\text{NO}_2$  a  $\text{C}_2\text{H}_2$ .

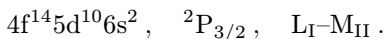


Jednodušší racionální a strukturní vzorce lze případně vysázet pomocí symbolu minus ( $-$ ), rovnítka ( $=$ ) a symbolu ekvivalence ( $\equiv$ ), které sázíme bez mezer. Takže například kyanid (sodný) lze vysázet jako  $\text{K}-\text{C}\equiv\text{N}$  (zdroják `\rm K{-}C{\equiv}N`) a fosgen jako  $\text{CCl}_2=\text{O}$  (zdroják `\rm CCl_2{=}O`), je to ale spíš pouze východisko z nouze. Sazba složitějších racionálních, strukturních či elektronových vzorců a chemických reakcí je netriviální problém, který řeší specializované knihovny maker a kterým se zde nebudu zabývat.

Chemické vzorce se samozřejmě sází antikvou i v indexech, takže např.  $\text{C}_{\text{H}_2\text{O}}$  by mohlo být označení tepelné kapacity vody, kdežto  $C_{\text{H}_2\text{O}}$  je paskvil.

Prakticky vše, co jsem uvedl o značkách chemických prvků, bych mohl zopakovat pro symboly různých (víceméně) elementárních částic:  $e$  (elektron),  $p$  (proton),  $n$  (neutron),  $u$ ,  $d$ ,  $s$ ,  $c$ ,  $b$ ,  $t$  (kvarky), a spousta dalších. Symbol pro elektronové neutrino  $\nu_e$  má tedy v indexu  $e$  a nikoli  $e$ .

Do této kategorie patří také označení elektronových slupek ( $K$ ,  $L$ , ...) a podslupek ( $s$ ,  $p$ ,  $d$ , ...) v atomech a všechny možné spektrální čáry a jejich série či elektronové konfigurace. Příklady:



Se symboly částic označovaných velkými řeckými písmeny (např.  $\Sigma^+$ ,  $\Xi^0$ ,  $\Omega$ ) je trochu potíže, protože  $\text{T}_\text{E}_\text{X}$  sází velká řecká písmena implicitně kolmým písmem (což chceme), pak je ale vhodné sázet proměnné písmem skloněným ( $\Sigma$ ,  $\Xi$ ,  $\Omega$ ). To zařídí příkaz `\mit`; skloněné velké sigma v předchozím příkladu jsem tedy vysázal `\mit\Sigma`. Pokud máme vhodné fonty, měli bychom odlišovat označení fotonu od proměnné  $\gamma$ , označení  $\mu$ -mezonu od proměnné  $\mu$ , apod.

## 1.6 Slova a části slov

Dalším typem objektu, který je třeba sázet základním řezem písma, jsou slova a části slov, které se ve vzorcích objevily „shodou okolností“. Přesněji, ve vzorcích se občas vyskytnou různá slova, slovní spojení nebo



části slov – protože se nejedná vůbec o symboly nějakých objektů, natož proměnných, je nesmysl sázet je matematickou italicou. Nejčastěji se objevují slova jako „pro“, „a“, „nebo“, „jinak“, „tj.“, „tedy“, „je-li“, apod. Tato slova je nejlépe vsadit do `\hboxu`, což zajistí, že se budou sázet stejným písmem, jako okolní text. Navíc uvnitř `\hboxu` fungují všechny „vymoženosti“ horizontálního módu, jako mezery a akcenty, takže je to lepší řešení než pouhé `\rm`. Píšeme tedy

$$|x+y| > |z| \quad \text{neboli} \quad |x| > |z| - |y|,$$

abychom dostali

$$|x + y| > |z| \quad \text{neboli} \quad |x| > |z| - |y|.$$

Toto se fakticky týká pouze `display` módu, protože u vzorců v odstavci je vždy rozumné opouštět matematický mód, když chceme napsat slovo (více se na toto téma dozvíte v kapitole Mezery v matematice).



Text v `\hboxu` uvnitř matematického módu `TEX` sází vždy aktuálním fontem nejbližšího vnějšího horizontálního módu. To může být v pořádku v případech, kdy se třeba rozhodneme-li sázet definice a věty skloněným písmem – pak v nich budou i slova uvnitř vzorců tímto písmem (použijeme-li pro ně `\hbox`). Na druhé straně bychom v případech jako

`$$ \sum_{\hbox{pro všechna } i \in M} $$`,

dostali „pro všechna“ vysázené základní velikostí písma, což by vypadalo dost obhludně. Zde musíme buď zmenšit písmo explicitně, nebo si nadefinovat příkaz `\mathbox`, který na to bude myslet za nás

```
\def\mathbox#1{\mathchoice{\mB\textfont{#1}}{\mB\textfont{#1}}
  {\mB\scriptfont{#1}}{\mB\scriptscriptfont{#1}}}
\def\mB#1#2{\hbox{\ifnum\fam<0 \fam=0 \fi \the#1\fam#2}}
```

Dalším elementárním příkladem jsou slova zastupující různé výrazy, jejichž přesný tvar není podstatný – snad nejčastěji používaná je zkratka „konst.“ označující, že je něco konstantní. Sázíme tedy

$$pV = \text{konst.}, \quad G(t) = 1 + t + 2t^2 + \text{členy vyšších řádů}$$

a nikoli

$$pV = \textit{konst.}, \quad G(t) = 1 + t + 2t^2 + \textit{členy vyšších řádů}.$$

## 1.7 Indexy a exponenty

Slova a části slov se hojně vyskytují také v indexech (příp. exponentech). U vícepísmenných indexů je obvykle zřejmé, kdy se jedná o část slova (a kdy o nějaké tenzorové indexy), takže snad nikdo nebude váhat použít v indexu antikvu v případech jako  $x_{\text{střední}}$ ,  $R_{\text{hvězdy}}$ ,  $A_{\text{max}}$ ,  $p_{\text{atm}}$ ,  $H_{\text{int}}$ ,

apod. (zdrojáky: `x_{\rm střední}`, `R_{\rm hvězdy}`, `A_{\rm max}`, `p_{\rm atm}`, `H_{\rm int}`).

V případě jednopísmenných indexů je k rozhodnutí, zda index sázet matematickou italicou či základním řezem písma, obvykle třeba znát kontext. Sážíme tedy  $C_p$  (`C_p`), vztahuje-li se ono  $p$  k proměnné  $p$  (např. tlaku); vztahuje-li se však k podložce, je správně  $C_p$  (`C_{\rm p}`). Stejně tak vysázíme  $n$ -tý člen posloupnosti jako  $a_n$  (`a_n`), kdežto jedná-li se o normálovou složku zrychlení  $a$ , vysázíme  $a_n$  (`a_{\rm n}`). A ještě jeden příklad, neboť opakování je matka zblbnutí: Symbol  $n_0$  (`n_0`) může označovat třeba počet králíků při nulovém počtu lišek, kdežto index lomu pro ordinární paprsek vysázíme  $n_o$  (`n_{\rm o}`), symbol  $n_o$  (`n_o`) se pak prakticky nevyskytuje (resp. neměl by), protože většina lidí má dost rozumu neoznačovat proměnné písmenem  $o$ .

## 1.8 Konstantní symboly

Tuto kategorii objektů bohužel neumím nějak stručně a jasně charakterizovat; naštěstí je jich tak málo, že je v podstatě mohu všechny vyjmenovat. Jsou to:

- písmeno  $e$  označující Eulerovo číslo;  $e = 2,718\ 281\dots$
- písmeno  $i$  označující imaginární jednotku;  $i^2 = -1$
- malé řecké písmeno  $\pi$ , označující Ludolfovo číslo;  $\pi = 3,141\ 592\dots$
- písmeno  $C$  označující Eulerovu konstantu (také se označuje malým řeckým  $\gamma$ );  $C = 0,577\ 215\dots$
- symbol  $id$  označující identitu – ten vyhovuje obecnému pravidlu, nepatří však do žádné z předchozích kategorií, takže ho tu uvádím explicitně

Možnost odlišit skloněné a kolmé malé řecké  $\pi$  (příp.  $\gamma$ ) v  $\text{T}_{\text{E}}\text{X}$ u standardně není, k dispozici jsou pouze skloněná malá řecká písmena (vhodné fonty nicméně existují, případně je lze vytvořit – výsledek jednoho mého pokusu je vidět níže). Zatímco u  $\pi$  si můžeme většinou dovolit neodlišovat skloněné a kolmé, protože se používá k označení něčeho jiného než Ludolfova čísla pouze řídce, písmena  $e$ ,  $i$  a  $C$  se běžně používají i k označení proměnných.

Sázíme tedy (pokud vzorec neznáte, upozorňuji, že  $J_0$  je zde Besselova funkce nultého řádu)

$$\int_0^{2\pi} e^{ia \cos x} dx = 2\pi J_0(a),$$

zdroják:

$$\int_0^{2\upi} e^{\im a\cos x}\,d x = 2\upi{\rm J}_0(a),$$

kde  $\mathcal{J}$ ,  $\mathcal{I}$  jsou definovány podle vzoru  $\mathcal{J}(\mathcal{J})$  a  $\mathcal{I}$  sází již zmiňované kolmé pí.

## 2 Mezery v matematice

Každý si už jistě všiml, že v matematických vzorcích  $\text{\TeX}$  zcela ignoruje mezery ve vstupním souboru a sází mezery po svém. Proto budou  $\$ x^3 + y = 5m \$$  a  $\$ x^3+y=5m\$$  vysázeny zcela stejně, a to jako  $x^3 + y = 5m$ .  $\text{\TeX}$  nám tím šetří spoustu práce (a chyb), protože navzdory tomu, že jeho automatická mezerovací pravidla jsou celkem jednoduchá, fungují ve většině standardních případů dobře.

$\text{\TeX}$  nicméně nerozumí významu toho, co píšeme, takže se nelze divit, že existuje řada celkem běžných situací, kdy automatické mezerování nefunguje. Klasickým případem syntaktické nerozhodnutelnosti jsou výrazy

$$n(n-1)m(m-1) \quad \text{a} \quad n(n-1)m(m-1).$$

První z nich ( $n(n-1)m(m-1)$ ) má význam součinu čtyř čísel  $n$ ,  $n-1$ ,  $m$  a  $m-1$ , kdežto druhý ( $n(n-1)\backslash,m(m-1)$ ) označuje součin funkce  $n$  s argumentem  $n-1$  a funkce  $m$  s argumentem  $m-1$ . Vidíme, že do druhého výrazu jsme museli dopsat explicitní mezeru, aby byl vysázen správně. Podobně tomu je ve většině případů, kde musíme automatické mezerování opravovat, protože  $\text{\TeX}$  prakticky nikdy nedělá mezery navíc.

Než se ale můžeme začít zabývat tím, kde a jak automatické mezerování opravovat, musíme vědět, jak funguje. Následuje proto trocha suché teorie.

### 2.1 Automatické mezerování

Upozorňuji čtenáře, že tento odstavec obsahuje řadu nepřesných či zjednodušujících tvrzení – podrobný popis automatického mezerování lze nalézt např. v kapitolách 17 a 18 a dodatku G  $\text{\TeX}$ booku či v kapitole 5  $\text{\TeX}$ booku naruby.

Při sazbě symbolů v matematickém módu vytváří  $\text{\TeX}$  dvanáct typů *atomů*, z nichž je pro účely této kapitoly podstatných osm:

- Ord – běžné (řádné, ordinární) matematické symboly ( $a$ ,  $b$ ,  $X$ ,  $\vartheta$ ,  $\infty$ )
- Op – (velké) unární matematické operátory ( $\sum$ ,  $f$ ,  $\cos$ ,  $\sup$ )
- Bin – binární operátory ( $+$ ,  $*$ ,  $\cap$ ,  $\vee$ )
- Rel – binární relace ( $=$ ,  $<$ ,  $\gg$ ,  $\notin$ )
- Open – otevírací závorky ( $\langle$ ,  $[$ )
- Close – zavírací závorky ( $\rangle$ ,  $]$ )
- Punct – interpunkce (čárka)
- Inner – výsledek sazby s `\left`, `\right`

Každý symbol má přitom v Plain  $\TeX$ u určeno, jakému druhu atomu má dát vzniknout. Pokud chceme některý symbol použít v jiném významu než defaultním, můžeme ke změně jeho typu využít příkazů `\mathord`, `\mathop`, `\mathbin`, `\mathrel`, `\mathopen`, `\mathclose`, `\mathpunct` a `\mathinner`, jejichž význam je zřejmý.

Typ atomu určuje, jak má být sázen.  $\TeX$  vkládá mezi atomy mezery podle následující tabulky, která vychází z ustálených konvencí matematické sazby

		<i>Pravý atom</i>							
		Ord	Op	Bin	Rel	Open	Close	Punct	Inner
<i>Levý atom</i>	Ord	0	1	2*	3*	0	0	0	1*
	Op	1	1	–	3*	0	0	0	1*
	Bin	2*	2*	–	–	2*	–	–	2*
	Rel	3*	3*	–	0	3*	0	0	3*
	Open	0	0	–	0	0	0	0	0
	Close	0	1	2*	3*	0	0	0	1*
	Punct	1*	1*	–	1*	1*	1*	1*	1*
	Inner	1*	1	2*	3*	1*	0	1*	1*

Čísla 0, 1, 2 a 3 v tabulce přitom znamenají po řadě žádnou, malou, střední a velkou mezeru – poslední tři jsou obvykle vytvářeny také příkazy `\,`, `\>` a `\;`. Hvězdička u čísla znamená, že mezeru není vkládána v indexech a exponentech. Pomlčka pak znamená, že tato situace nemůže nastat, neboť atomy typu Bin mají tendenci proměňovat se v atomy typu Ord, je-li to třeba. To nám umožňuje např. psát `x=-3` a dostat  $x = -3$ , a nikoli něco jako  $x = -3$ , nebo psát relaci přiřazení prostě `i:=5` a dostat  $i := 5$ . Je třeba podotknout, že  $\TeX$  vždy vkládá mezery *mezi* nějaké atomy, nikdy tedy nevloží mezeru před první atom nebo za poslední atom (na jeho „vnitřní straně“ přitom mezeru vložit může).

Posledním pravidlem, které uvedu, je, že ze skupiny – tedy ze vstupu `\langle matematický materiál \rangle` – vytváří  $\TeX$  vždy atom typu Ord obsahující `\langle matematický materiál \rangle`. To vysvětluje, proč se lze mezery po čárce, která je normálně typu Punct, zbavit jednoduchou konstrukcí `\{, \}` a není nutné psát `\mathord, .`

## 2.2 Operátory, funkce a jejich argumenty

Nejčastěji musíme přidávat mezery mezi funkce či operátory vyskytující se v součinu z důvodu oddělení jejich argumentu od následujících členů.

Následující výčet si nečiní žádný nárok na úplnost, obsahuje pouze některé nejběžnější případy. Čtenář jistě dokáže aplikovat uvedené zásady na jiné podobné operátory a funkce. Na všechna uvedená tvrzení se přitom vztahuje pravidlo, že mezery vkládáme pouze mezi jednotlivé členy součinu, nikoli kolem celého výrazu.

### Pravá závorka

Za pravou závorkou ukončující argument funkce sázíme malou mezeru

$$u'(x)v(x) - u(x)v'(x), \quad \int f(x)g(x) dx.$$

Zdrojáky:

$$u'(x)\ ,\ v(x)-u(x)\ ,\ v'(x), \quad \int f(x)\ ,\ g(x)\ ,\ {\rm d}x.$$

### Jméno funkce

Malou mezeru vkládáme také před jméno obecné funkce je-li násobena proměnnou (před standardní funkce, které jsou typu Op, sází T<sub>E</sub>X mezeru automaticky). Funkce je také  $e^x$

$$xy'(x) + 2y(x) = xe^{-x} + 2\sin x.$$

$$\text{Zdroják: } x\ ,\ p'(x) + 2q(x) = x\ ,\ \, e^{-x} + 2\sin x.$$

V zájmu konzistentnosti můžete doplnit mezeru i do součinu  $2y(x)$ , nicméně číselné konstanty (zde dvojka) jsou již dostatečně odlišeny typem písma, takže to není nutné.

### Argument funkce

Je třeba vždy jasně odlišit, kde končí argument funkce. To se zejména týká následujících výrazů

$$\ln x(\ln \ln x)^2 \equiv \ln[x(\ln \ln x)^2], \quad \ln x(\ln \ln x)^2 \equiv (\ln x)(\ln \ln x)^2.$$

Máme-li jen trochu zdravého rozumu, vyhneme se samozřejmě zápisům, jež by mohly být snadno špatně pochopeny, úplně a raději explicitně závorkujeme.

### Odmocnina

V případech, kdy by mohlo dojít k přehlédnutí, sázíme malou mezeru za odmocninou

$$\sqrt{2}a, \quad \sqrt{\frac{p}{2} + qr}, \quad \sqrt{\sin xy}z.$$

$$\text{Zdrojáky: } \sqrt{2}\ ,\ a, \quad \sqrt{\{p\over 2\}+q}\ ,\ r, \quad \sqrt{\{\sin xy\}}\ ,\ z.$$

## Faktoriál

Pro přehlednost sázíme malou mezeru za faktoriálem

$$n! m! (m+n)!, \quad \frac{60!}{11! 13! 15!}.$$

Zdrojáky: `n!\,m!\,(m+n)!, {6!\over1!\,3!\,5!}`.

## Diferenciál

Před diferenciálem i za koncem jeho argumentu sázíme malou mezeru (podle starých norem by se měla sázet mezera i po diferenciálu – podle mne je to nerozum a nikdo to tak nedělá)

$$x dy - y dx, \quad r^2 \sin \psi dr d\psi d\phi.$$

Zdrojáky: `x\,dy-y\,dx, r^2\sin\psi\,dr\,d\psi\,d\phi`.

## Parciální derivace

Stejně jako u diferenciálu sázíme malou mezeru před symbolem parciální derivace a za koncem jeho argumentu

$$\frac{\partial^2 M}{\partial x \partial y} = \frac{\partial^2 M}{\partial y \partial x}.$$

Zdroják (`\p` je aliasem `\partial`):

$$\{\p^2 M \over \p x, \p y\} = \{\p^2 M \over \p y, \p x\}.$$

## Operátory $\nabla$ , $\square$ , $\Delta$ , $\delta$ a $\triangle$

Jsou-li symboly  $\nabla$ ,  $\square$ ,  $\Delta$ ,  $\delta$  a  $\triangle$  použity jako operátory, sázíme malou mezeru za jejich argumenty

$$\Delta \Gamma = \Delta n_1 \Delta n_2 \Delta n_3, \quad \nabla(a \nabla b) = \nabla a \nabla b + a \Delta b.$$

Zdrojáky:

$$\Delta \Gamma = \Delta n_1, \Delta n_2, \Delta n_3, \\ \nabla(a \nabla b) = \nabla a, \nabla b + a \Delta b.$$

Používáme-li symbol  $\Delta$  jako operátor, je vhodné používat pro proměnné raději jeho skloněnou variantu  $\triangle$ , kterou vysázíme `\mit\Delta`.

## Rovné závorky

Vyskytují-li se vedle sebe v součinu členy ohraničené rovnými závorkami `| a ||`, sázíme mezi nimi malou mezeru

$$|z| = |x| |y|, \quad \|\mathbf{uv}\| \leq \|\mathbf{u}\| \|\mathbf{v}\|.$$

Zdrojáky: `|z|=|x|\,|y|, \|\vec{uv}\|\le\|\vec{u}\|\,\|\vec{v}\|`.

## Kvantifikátory

Malou mezeru sázíme také za argumentem kvantifikátorů

$$(\forall x \forall y \forall z)(x^2 + y^2 + z^2 \geq 0).$$

Zdroják:

$$(\forall \text{forall } x \backslash, \backslash \text{forall } y \backslash, \backslash \text{forall } z)(x^2 + y^2 + z^2 \geq 0).$$

## 2.3 Interakce vzorců s větami

Sázíme-li v display módu na jeden řádek více vzorců, vkládáme mezi ně mezeru o velikosti čtverčíku, kterou vysázíme příkazem `\quad` (občas se sází i větší mezery, třeba 2 em (`\qquad`), nicméně v celém textu stejné)

$$a + b \geq c, \quad b + c \geq a, \quad c + a \geq b.$$

Stejnou mezeru vkládáme i mezi spojky (či jiná spojující slova) a jednotlivé vzorce

$$|x + y| < z \quad \text{a} \quad |y + z| < x.$$

Tato slova a části slov umísťujeme do `\hboxu`, protože v něm fungují všechny vymoženosti horizontálního módu jako např. mezislovní mezery  $x = 0$  takžesenikdy nedopustíte metétoosudové chyby  $y = 0$ .

Pokud za vzorcem následuje interpunkce (patřící do věty, ne do vzorce), sázíme mezi ni a vzorec mezeru, jak je vidět i ze všech příkladů v tomto Bestiáři. Nevím bohužel, jak velkou mezeru přesně správně sázet – v Bestiáři používám mezeru `\:` o velikosti 4 mu, ale je možná i menší, např. `\,` o velikosti 3 mu.

Je třeba důsledně odlišovat mezery po interpunkci v matematických výčtech a mezislovní mezery v normálních větných výčtech. Srovnejte například mezery v následujících příkladech

$$\text{množina } M = \{0, 1, a, b, c\}$$

nezáporná čísla  $0, 1, a, b$  a  $c$

První příklad byl vysázen `\$M = \{0, 1, a, b, c\}\$`, kdežto druhý `0, ~1, $a$, $b$ a ~$c$`. Rozdíl by se přitom ještě zvětšil, kdyby byla proměnná `\mathsurround`, jejíž hodnota určuje velikost mezery vkládané kolem vzorců, nastavena na nenulovou hodnotu.

Dalším dobrým důvodem, proč každou proměnnou ve výčtu umísťovat do vlastních `\$`, je lámání řádků: `\TeX` nikdy nezlomí seznam `\$x, y, z, w\$`, zatímco seznam `\$x$, ~$y$, $z$ a ~$w$` může být zlomen na kterémkoli vhodném místě (všimněte si umístění vlnek).



## 2.4 Symboly, které se pletou

Prakticky každý matematický symbol se používá v několika různých významech. Většinou se přitom ale nemění jeho typ. Existuje však několik symbolů, které se běžně používají jako různé typy lišící se mezerováním.

### Čárka

Problémy s čárkou jsou podrobně rozebrány v kapitole Sazba číselných hodnot, takže jen stručně: čárku ve významu interpunkce sázíme `,`, kdežto ve významu desetinné čárky jako `{,}`.

### Tečka

Používá se ve významu desetinné tečky, jako binární operátor a interpunkce

$$\pi = 3.145\,926\,535\dots, \quad 3 \cdot 2^x - 2 \cdot 2^{x-1} = 2^{x+1}.$$

Tečku ve významu desetinné tečky sázíme `.`, jako typ `Punct` ji sází příkaz `\ldotp`, ten se nicméně nikdy přímo nepoužívá, protože v matematických vzorcích se tečka ve významu interpunkce vyskytuje výhradně v elipse, kterou sázíme `\ldots`. Příkaz pro tečku typu `Bin` v `Plainu` definován není, já například používám příkaz `\.` definovaný v úvodním seznamu `maker`. Upozorňuji, že ještě existuje příkaz `\dot`, který nemá s uvedenými významy tečky nic společného a sází akcent  $\dot{a}$ .

Důvody, proč se v české typografii používá pro násobení tečka na řádku a nikoli centrovaná (`\cdot`), jsou ovšem spíše historické, a pokud se vám více líbí centrovaná tečka, použijte raději ji

$$7 \cdot 11 \cdot 13 = 1001.$$

To, že obě varianty jsou akceptovatelné, samozřejmě neznamená, že byste je mohli míchat.

### Dvojtečka

Používá se jako binární operátor, relace a interpunkce

$$2 + (6 : 3) = 4, \quad \text{broskvoň} : \text{broskev} = ??? : \text{kaštan},$$

$$\varphi: A \rightarrow B, \quad M = \{x \in U: V(x)\}.$$

`Plain TEX` nastavuje dvojtečce typ `Rel`, takže v prvním vzorci stačí psát `:`. Pro dvojtečku jako `Punct` musíme ve třetím a čtvrtém vzorci použít příkaz `\colon` (nebo explicitní `\mathpunct:`); navíc je mezera 3 mu za `Punct` v tomto případě nedostatečná, takže píšeme `g\colon A\to B`, `x\in U\colon V(x)`. Pro dvojtečku typu `Bin` žádný příkaz předdefinovaný není, použijeme `\mathbin:` nebo si nějaký nadefinujeme.

V řadě případů není bohužel vůbec jasné, zda je dvojtečka použita ve smyslu naznačeného dělení (tj. relace poměru) nebo provedeného dělení (tj. operátoru). I když pro operátor dělení existují i jiné symboly ( $/$  a  $\div$ , druhý z nich se ovšem v češtině nepoužívá), ne vždy se nám podaří se uvedeným nejednoznačností vyhnout.



Pokud bychom chtěli používat dvojtečku převážně jako typ Bin, můžeme změnit její typ globálně (a případně dodefinovat příkaz pro dvojtečku typu Rel)

```
\mathcode'\:="203A
```

## Svislá čára |

Používá se jako relace (např. dělitelnosti) a při udávání prvků množiny, kdy ji sázíme jako `\mid`

$$16 \mid 7^{2k} - 1, \quad M = \{x \in U \mid V(x)\}.$$

Ve významu symbolu pro velikost (mohutnost, absolutní hodnotu, ...), tj. rovné závorky

$$|y| \geq |x - y| - |x|, \quad |-r| = \left| \frac{p}{q} \right|$$

ji většinou stačí sázet jako | (též dosáhneme příkazem `\vert`), protože jí Plain T<sub>E</sub>X přiřazuje typ Ord. V druhém příkladu jsme nicméně museli použít `\left|` a `\right|`

```
\left|-r\right| = \left|\frac{p}{q}\right|,
```

protože jinak by si minus rychle vzpomnělo, že je vlastně typu Bin, a dostali bychom obudné  $|-r|$ . Přetypování na otevírací a zavírací závorku pomocí `\mathopen` a `\mathclose` by bylo také možné, ale `\left` a `\right` zvládnou tuto práci stejně dobře a navíc získáme závorky, jejichž velikost se sama přizpůsobuje.

## Dvojitá svislá čára ||

Používá se jako relace (např. rovnoběžnosti) a jako symbol pro normu

$$\mathbf{a} \times \mathbf{b} \parallel \mathbf{c} \quad \text{a} \quad \|x\| \geq 0.$$

Dvojitou svislou čáru jako typ Ord vysází příkaz `\|` nebo `\Vert`, jako typ Rel příkaz `\parallel`. Stejně jako u jednoduché svislé čáry je v případě jejího použití jako závorky občas potřeba ji na závorku skutečně přetypovat pomocí `\left` a `\right`.

## Písmena $\Sigma$ a $\Pi$

Velká řecká písmena sigma a pí se používají i jako symboly pro sumaci a součin – tyto symboly se nicméně liší nejen mezerováním, ale i velikostí

(a umístěním indexů)

$$\Sigma_1 + \Pi_2 + \Omega_0 = \Xi, \quad S = \sum_{i=1}^N \prod_{k=1}^i (x - x_k).$$

Velká řecká písmena sigma a pí vysázíme příkazy `\Sigma` a `\Pi` (jsou typu Ord), symboly pro sumaci a součin příkazy `\sum` a `\prod` (jsou typu Op). Na druhé straně, pokud se sigma objeví např. v indexu k označení toho, že se něco pouze vztahuje k sumaci (a nejedná se tedy přímo o operátor), použijeme normálně `\Sigma`

$$y_1 = f(x_1), \quad y_2 = f(x_2), \quad \dots, \quad y_N = f(x_N), \quad y_\Sigma = f\left(\sum_{i=1}^N x_i\right).$$

### Hranaté závorky [ , ]

Používají se vždy jako závorky, ale v poslední době se rozšiřuje označování otevřených intervalů *opačnými* závorkami

$$x \in ]-a, a[, \quad (p, q) \in [0, +\infty[ \times ]-\infty, 0]$$

namísto u nás tradičního

$$x \in (-a, a), \quad (p, q) \in \langle 0, +\infty \rangle \times \langle -\infty, 0 \rangle.$$

Nepovažuji to za příliš šťastný nápad, ale když už budeme takové otočené závorky sázet, musíme je vždy přetypovat: `\mathopen]` a `\mathclose[`.

### Lomené závorky ⟨ , ⟩

Tyto závorky se doufám používají pouze jako závorky, řada lidí si je nicméně plete se symboly nerovností `<` a `>`, které se liší nejen mezerováním, ale i tvarem. Vznikají potom následující paskvily

$$A = \langle 1; 3; -2 \rangle, \quad w \in \langle a, +\infty \rangle, \quad \langle x \rangle = \int x \varrho(x) dx.$$

Lomené závorky vysázíme příkazy `\langle \rangle` a `\rangle \langle`, čímž uvedené paskvily tím spravíme na

$$A = \langle 1; 3; -2 \rangle, \quad w \in \langle a, +\infty \rangle, \quad \langle x \rangle = \int x \varrho(x) dx,$$

zdrojáky: `A = \langle 1; 3; -2 \rangle`, `w \in \langle a, +\infty \rangle`, `\langle x \rangle = \int x \varrho(x) dx`.



V případě lomených závorek a svislé čáry je situace komplikovaná jejich použitím v kvantové mechanice ve smyslu tzv. bra a ket vektorů, které se obvykle sází

$$\langle a | \hat{B} + \hat{A} | a' \rangle = \langle a | \hat{B} | a' \rangle + a' \langle a | a' \rangle.$$

Problém sazby bra a ket vektorů však přesahuje rámec tohoto Bestiáře, takže pouze upozorním na moji knihovnu `bracketsym`, která ho celkem uspokojivě řeší.

A aby těch trablů se symboly nerovností nebylo málo, pokouší se pomocí nich občas někdo vytvořit dokonce i symboly řádových nerovností  $\ll$  a  $\gg$ , čímž dostává obludy typu

$$a \gg b, \quad \varepsilon \ll 1.$$

Symboly  $\ll$  a  $\gg$  sázíme jako `\ll` a `\gg` (v české typografii je zvykem používat „uříznuté“ varianty, ty ale nejsou běžně k dispozici).

## 2.5 Mezery z estetických důvodů

V některých případech je možné (vhodné) upravit mezerování z čistě estetických důvodů. Příčinou je obvykle to, že v matematickém módu nefunguje obvyklý kerning, takže některé symboly občas vyjdou trochu příliš daleko od sebe (případně příliš blízko k sobě).

K přisunutí symbolů blíže k sobě používáme zápornou malou mezeru `\!`, nejčastějšími kandidáty na její použití jsou lomítko, integrál a různé nesymetrické symboly. Srovnejte

$$x^7/7, \quad n/\log n, \quad \Gamma_2 + \Lambda^2, \quad \iiint_D dx dy$$

s neupravenými výrazy

$$x^7/7, \quad n/\log n, \quad \Gamma_2 + \Lambda^2, \quad \int \int_D dx dy.$$

První řádek příkladu přitom byl vysázen

$$\begin{aligned} &x^7\!/7, \\ &n/\!\log n, \\ &\Gamma_{\!2} + \Lambda^{\!2}, \\ &\int\!\!\!\!\!\int\!\!\!\!\!\int\!_D \! \, dx \!, \! dy. \end{aligned}$$

Občas můžeme upotřebit i (kladnou) malou mezeru `\,`. Srovnejte

$$[0, 1), \quad \sqrt{\ln n}$$

s neupravenými výrazy

$$[0, 1), \quad \sqrt{\ln n}.$$

První vztahy byly přitom vysázeny `[\,0,1)`, `\sqrt{\, \ln n}`.

Kandidáty na opravy jsou také kulaté závorky větších velikostí. Srovnejte

$$\left(\frac{1}{y} + \frac{2}{y}\right)^3, \quad \Gamma(x)\Gamma\left(x + \frac{1}{2}\right).$$

s neupravenými výrazy

$$\left(\frac{1}{y} + \frac{2}{y}\right)^3, \quad \Gamma(x)\Gamma\left(x + \frac{1}{2}\right)$$

První řádek byl přitom vysázen

```
\(\frac{1+y}{y}\)^{\!\!3},  
\Gamma(x)\!,\Gamma\!\!(\!x+\frac{1}{2}\!)
```

Dlužno podotknout, že na vině jsou částečně ony automatické závorky `\(` a `\)`; kdybychom velikosti závorek nastavili ručně `\biggl(` a `\biggr)`, nebylo by oprav potřeba tolik. Oba přístupy mají svoje nevýhody.

## 3 Sazba číselných hodnot

Tato kapitola se zabývá téměř výhradně sazbou čísel v českých textech technického charakteru. Sazby čísel v textech netechnického charakteru se týkají odstavce 3.1 (Obecné zásady), 3.6 (Procenta, promile, stupně, grady a jiné speciality), 3.7 (Peněžní hodnoty) a 3.8 (Časové údaje a kalendářní data). Sazbou číselných hodnot podle anglosaských konvencí se pak zabývá odstavec 3.5.

### 3.1 Obecné zásady

Hlavní zásada je prostá: nesázet čísla matoucím způsobem. To znamená zejména:

- čísla se nikdy nedělí (přesněji, čísla je možné dělit, ale je to složité – viz kapitolu Lámání vzorců)
- čísla se nikdy neoddělují od jednotek a nedělí se ani časové údaje vyjádřené číslem
- s výjimkou časových údajů, skutečně velkých čísel a čísel spojených s měrnými jednotkami by se měla čísla sázet výhradně slovy – takže žádné „mám 3 psy“ nebo dokonce „jdeme na 1 pivo“
- řádek nemá začínat číslem – je třeba přeskupit text, nebo hodnotu vypsát slovy (v případě technických textů je třeba toto pravidlo občas brát trochu volněji)
- normální věty nikdy nezačínají číslem
- paskvily jako „v 65-ti letech“, „273-krát“ či „151-ní strom“ jsou chybami gramatickými; sázíme „v 65 letech“, „273krát“ (resp. „273×“, kde to  $\times$  není malé „x“, ale  $\times$ ) či „151. strom“ – jiná věc je, jde-li o proměnnou, zápisy  $2m$ -úhelník,  $(k+1)$ -tý či  $n$ -tice jsou v pořádku (pokud už se jim nemůžeme vyhnout)
- označujeme-li poznámky pod čarou čísla, nesmíme je umísťovat tak, že by mohly být zaměněny s mocninou
- nikdy za sebe nesázíme dvě čísla oddělená pouze mezerou, obraty typu „zisk činil v roce 96 664 mil. korun“ či „zatímco v hale č. 8 615 vadných součástí“ je třeba přeformulovat

Nepíšeme tedy věci jako

...potřebujeme 60 dkg čerstvých listů indického konopí...

protože tím mezi číslem a jednotkou vznikne mezerová mezera, ve které může dojít k řádkovému zlomu. V kuchařce, kde se žádné složité jednotky nevyskytují, je použitelná jednoduchá oprava na `potřebujeme~60~dkg`. Musíme přitom doufat, že nám  $\text{\TeX}$  najde dobrá místa řádkového zlomu

i s tolika vlnkami, jinak nezbyvá než přeformulovat text nebo šedesát vypsát slovy. V textech technického charakteru je ale třeba řešit věci systémověji.

## 3.2 Úvodní příklad

Nyní se vrhněme na ony texty technického charakteru, tedy na pěkně složitá čísla, a ještě navíc s jednotkami.

Zápis číselné hodnoty nějaké veličiny se svým vzhledem celkem dost liší od „průměrného matematického vzorce“. Nelze se tedy příliš divit, že když  $\text{\TeX}$  necháme všechny symboly vysázet implicitním způsobem, bude výsledek ubohý. Jako příklad uvedu hodnotu Planckovy konstanty, kterou bychom se mohli pokusit vysázet tak, že prostě napíšeme

```
\hbar=1,0545887.10^{-34} kg.m^2.s^{-1}.
```

Výsledkem ovšem bude následující obluda:

$$\hbar = 1,0545887.10^{-34}kg.m^2.s^{-1} .$$

Téměř celý zbytek této kapitoly se bude zabývat tím, jak to spravit, tj. jak dosáhnout něčeho jako

$$\hbar = 1,054\ 588\ 7 . 10^{-34} \text{ kg m}^2 \text{ s}^{-1} ,$$

resp.

$$\hbar = 1,054\ 588\ 7 . 10^{-34} \text{ kg} . \text{ m}^2 . \text{ s}^{-1} ,$$

což by měla být hodnota Planckovy konstanty vysázená podle všech (českých) typografických konvencí.

## 3.3 Písmo jednotek

Patrně nejnápadnější chybou je písmo jednotek, které se mají sázet základním řezem písma (antikvou). Platí to pro všechny jednotky bez výjimky, včetně předpon pro násobné a dílčí jednotky (k pro kilo-, m pro mili-, n pro nano-, G pro giga-, atd.), které se neoddělují mezerou od jednotky: kg, cm, pF, MWh, GeV. Pokud máme k dispozici příslušný řez písma, měli bychom sázet kolmým písmem i  $\mu$  označující předponu mikro-.

Písmo opravíme snadno dopsáním  $\text{\rm}$  před jednotky (opustit pro sazbu jednotek matematický mód je také možné, ale dost si tím zkomplikujeme sazbu případných mocnin)

```
\hbar=1,0545887.10^{-34}\rm kg.m^2.s^{-1},
```

tím se vzhled trochu zlepší na

$$\hbar = 1,0545887 \cdot 10^{-34} \text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1} .$$



Různé exotické (a zastaralé) jednotky se sází různými exotickými způsoby (přestože vždy antikvou) a je dobré mít nějaký důvěryhodný pramen, chceme-li správně vysázet věci jako  $\text{lbf in}^{-2}$  (libra síly na čtverečný palec),  $\text{cal}_{\text{th}}$  (termochemická kalorie) či  $\text{mmH}_2\text{O}$  (milimetr rtuťového sloupce).

### 3.4 Mezery

Chybějících či přebývajících mezer budeme muset spravit celkem hodně.

#### Tečka jako binární operátor

Předně musíme zařídit, aby se tečka, která se zde používá ve smyslu operátoru násobení a kterou Plain  $\text{T}_{\text{E}}\text{X}$  implicitně sází jako typ `Ord`, sázela jako binární operátor – a nebo použít centrovanou tečku `\cdot`. Nebudeme ovšem dopisovat `\mathbin` před každou tečku a raději použijeme příkaz `\.` z úvodního seznamu `maker`.

Potom můžeme psát

$$\hbar=1,0545887\cdot 10^{-34}\text{kg}\cdot\text{m}^2\cdot\text{s}^{-1},$$

což  $\text{T}_{\text{E}}\text{X}$  vysází

$$\hbar = 1,0545887 \cdot 10^{-34} \text{kg} \cdot \text{m}^2 \cdot \text{s}^{-1} .$$

Jinou možností, jak se vypořádat s operátory násobení mezi jednotkami, je sázet namísto nich pouze úzké mezery `\.`, jako to dělají Angličané

$$\hbar = 1,0545887 \cdot 10^{-34} \text{kg m}^2 \text{s}^{-1} .$$

Podle mne je tato volba velmi rozumná. V každém případě musíme být konzistentní a nesázet mezi jednotky pokaždé něco jiného.

#### Mezera mezi číslem a jednotkami

Dále doplníme mezeru mezi číslo a první jednotku. Názory na velikost této mezery se různí, ale mezerou `\:` o velikosti 4 mu (nejmenší mezi-slovní) nic nezkažíme. Vylepšujeme tedy zdroják na

$$\hbar=1,0545887\cdot 10^{-34}\:\text{kg}\cdot\text{m}^2\cdot\text{s}^{-1}$$

a dostáváme

$$\hbar = 1,0545887 \cdot 10^{-34} \text{kg m}^2 \text{s}^{-1} .$$

#### Mezera za desetinnou čárkou

Za desetinnou čárkou naopak mezera nemá co dělat. Čárku sází Plain  $\text{T}_{\text{E}}\text{X}$  implicitně jako interpunkci, takže ji musíme přetypovat na ordinární



symbol. Mohli bychom to udělat příkazem `\mathord`, elegantnější ale je uzavřít ji do skupinových (složených) závorek (viz kapitolu Mezery v matematice)

```
\hbar=1{,}0545887\cdot 10^{-34}\:\rm kg\,m^2\,s^{-1}.
```

Výsledek je už celkem uspokojivý

$$\hbar = 1,0545887 \cdot 10^{-34} \text{ kg m}^2 \text{ s}^{-1}.$$

Nejjednodušším řešením problémů s čárkou se zdá být používat namísto ní po anglosaském způsobu desetinnou tečku, za ní totiž  $\TeX$  mezeru nedělá. V českých zemích je však zvykem používat desetinnou čárku, a tak se jí držíme. A ať už budeme sázet čísla jakkoli, je základním pravidlem *nikdy nemíchat různé způsoby*.



Čárku můžeme také přetypovat na ordinární symbol globálně

```
\mathcode'\,="013B,
```

potom ji ovšem nelze jednoduše používat v (matematických) výčtech, jako například  $A = \{1, 4, 9, 16, \dots, n^2\}$ , což normálně dává správně vysázený seznam  $A = \{1, 4, 9, 16, \dots, n^2\}$ . Po přetypování bychom ovšem dostali příliš malé (tj. žádné) mezery po čárkách  $A = \{1,4,9,16, \dots, n^2\}$ .



Jinou možností je psaní desetinných teček, s tím, že tečky nastavíme `\mathcode` čárky typu Ord

```
\mathchardef\period=\mathcode'.
```

```
\mathchardef'\,="013B
```

Tím se ovšem promění na čárky *úplně všechny* tečky ve vzorcích, tedy i ty, které se tam objevily jako větná interpunkce, takže je musíme psát jako `\period`, což je asi komplikace, která nestojí za to.

## Mezery po třech řádech

K dokonalosti už teď chybí pouze jediné – malé mezery oddělující cifry po třech řádech. Doplňme po třech řádech malé mezery `\:`

```
\hbar=1{,}054\:588\:\cdot 10^{-34}\:\rm kg\,m^2\,s^{-1}
```

a máme přesně to, čeho jsme chtěli dosáhnout

$$\hbar = 1,054\ 588\ 7 \cdot 10^{-34} \text{ kg m}^2 \text{ s}^{-1}.$$

Uvedené malé mezery se nesází v případě, že je číslo pouze čtyřciferné, což se samozřejmě také týká letopočtů.



Mezery po třech řádech by normálně měly být na šířku čárky, která je ale v  $\TeX$ u (přesněji v CM fontech) větší (5 mu) než nejmenší mezislovní mezera, takže mezera 4 mu se zdá být rozumným kompromisem.

Telefonní čísla, PSČ a jiné výstřednosti mají ještě svoje vlastní pravidla;

v zásadě se sází ve skupinách po dvou a třech číslicích oddělených úzkou pevnou mezerou \:

PSČ 547 01, 05/43 32 77 88, 77 22 11-2.

### 3.5 Anglosaské konvence

Nezřídka kdy se nám asi přihodí, že budeme muset vysázet čísla podle anglosaských konvencí. Pokud je mi známo, dopadla by podle nich Planckova konstanta takto

$$\hbar = 1.054\,588\,7 \times 10^{-34} \text{ kg m}^2 \text{ s}^{-1} .$$

Vidíme, že se změnilly dvě věci. Jednak se desetinná čárka proměnila na tečku – tím jsme si ušetřili práci s přetyčováním. A dále se změnilly operátory násobení – mezi jednotky Angličané nepíší „nic“ (tj. pouze úzkou mezeru \,) a před  $10^{\text{něco}}$  píší křížek  $\times$  (\times). Zdroják tohoto příkladu je tedy

```
\hbar=1.054\:588\:7\times10^{-34}\:\rm kg\,m^2,s^{-1}.
```

### 3.6 Procenta, promile, stupně, grady a jiné speciality

Úvodem raději uvedu, že procento % v  $\TeX$ u obvykle označuje komentář. Chceme-li tedy mít % na výstupu, musíme psát \%. Symbol pro promile ‰ vysázíme \promile (v případě nouze \char141), symbol pro stupeň  $^\circ$  a symboly pro úhlovou minutu a vteřinu ' a '' . Poslední tři přitom lze použít pouze v matematickém módu (přesněji, ' a '' v odstavcovém módu také „něco“ vysází, ale namísto 12'45'' bychom dostali paskvil 12'45"). Značku pro stupeň nelze zaměňovat s malým „o“ v horním indexu.

Procenta a promile se sází jako každé jiné jednotky s jednou výjimkou: má-li číselný údaj funkci přídavného jména, přisazuje se k číslu bez mezery. Příklad:

Meziroční inflace je 22,5%. Vypil jsem 3% roztok kyanidu sodného.

Klíčová místa byla vysázena: 22,5\:% a 3\%.

Ani se značkou pro stupeň to není jednoduché. Ve významu teplotního stupně tvoří jednotku dohromady s následujícím písmenem (v našich krajích obvykle C, stupeň Celsia), ke kterému se tedy přisazuje bez mezery. Před ním se sází mezera obvyklá před jednotkami, úhlové míry se naproti tomu sází celé bez mezer. Příklady:

6°32'45'' , 3 °C , 420 °F

byly vysázeny:  $6^\circ 32' 45''$ ,  $3^\circ\text{C}$ ,  $420^\circ\text{C}$ . Příkaz  $\circ$  pro značku stupně Celsia přitom definujeme

$\def\circ{\text{\char"0000}^\circ\mkern-2mu\text{C}}$ ,

protože písmeno C je vzhledem k jeho tvaru třeba přisadit ke značce stupně trochu těsněji.

Grady, centigrady a setiny centigradu už doufám nikdo jako úhlovou míru nepoužívá, sází se nicméně do horního indexu a přisazují se bez mezer, jako všechny ostatní úhlové jednotky. Příklad:

$$\beta = 2^g 64^c 25^{cc},$$

zdroják:  $\beta = 2^{\text{g}} 64^{\text{c}} 25^{\text{cc}}$ .

Astronomové do toho celého ještě vnáší zmatek tím, že používají časové jednotky pro úhlové veličiny (konkrétně rektascenzi) a do horního indexu píší i jednotky hvězdné velikosti. Sázíme je stejně jako výše uvedené grady

$$\alpha = 11^h 26^{\text{min}} 48^s, \quad \Delta\alpha = -22,30^s, \quad m = -26,73^m,$$

zdrojáky:

$\alpha = 11^{\text{h}} 26^{\text{min}} 48^{\text{s}}$ ,

$\Delta\alpha = -22,30^{\text{s}}$ ,

$m = -26,73^{\text{m}}$ .

### 3.7 Peněžní hodnoty

Pro peněžní hodnoty platí stejná pravidla jako pro jiná čísla s jednotkami, ale je tu několik zvláštností. Jednak se u celých čísel používá pomlčka namísto desetinných nul, a dále se „jednotky“ často sází nejen za hodnotu ale i před ni. Příklady:

Kč 200,-, Kč 125,50, \$ 100, 80 Kč

byly vysázeny:  $\text{Kč}\backslash:200,-$ ,  $\text{Kč}\backslash:125,50$ ,  $\text{\$}\backslash:100$ ,  $80\backslash:\text{Kč}$ .

### 3.8 Časové údaje a kalendářní data

Časové údaje se v zásadě sází dvěma způsoby. Jednak tak, že mezi hodiny a minuty klademe tečku a mezi minuty a sekundy dvojtečku, obojí bez mezer (nicméně např. při sazbě sportovních výsledků se oddělují dvojtečkou i hodiny a minuty). Příklad:

16.40:32,2.

Druhou možností je sázet je prakticky stejně jako každé jiné jednotky

(tedy se zúženou mezerou)

16 h 40 min 32,2 s.

V časových intervalech se sází pomlčka (*en-dash*), a to *bez mezer*, stejně jako v jakýchkoli jiných výčtech. Příklad:

10.00–11.30 h.

Zdroják: 10.00--11.30\,h.

Kalendářní datum se sází řadou způsobů (což ale není totéž, jako jak nás zrovna napadne). Den se sází arabskými číslicemi s tečkou. Měsíc se sází arabskými (řídce též římskými) číslicemi s tečkou a nebo slovy. Letopočet se sází arabskými číslicemi, v textech speciálního charakteru i římskými. Jsou-li vyjádřeny číslem, oddělují se jednotlivé údaje zúženými mezerami.

Pokud se jedná o zkracování letopočtu o první dvě číslice, tak hlavní zásadou je nezkracovat, což se zejména týká letopočtů po roce 2000. Zkracování je možné v názvech akcí obsahujících ročník (např. Setkání 99), názvem akce pak neoddelujeme od ročníku, píšeme tedy *Setkání~99*. Dále je zkracování možné, vyskytují-li se vedle sebe dva letopočty (ze stejného století) oddělené pomlčkou či lomítkem. Příklady

12. 3. 1997, 22. listopadu 2011, 8. VIII. 1994, Setkání '99,  
v letech 1987–90, školní rok 1998/99,

byly vysázeny

12.\,3.\B\,1997,  
22.\,listopadu 2011,  
8.\,VIII.\B\,1994,  
Setkání~'99,  
v~letech 1987--90,  
školní rok 1998/99.

Časové údaje se nikdy nedělí, kalendářní data nanejvýš mezi měsícem a letopočtem.

## Licence

Yetiho typografický bestiář. Copyright © 1997–2003 David Nečas (Yeti).

Tento dokument je svobodný dokument. Můžete ho šířit a/nebo volně upravovat v souladu s verzí 2 nebo vyšší licence GNU Library General Public License publikovanou nadací Free Software Foundation.

Tento dokument je šířen v naději, že bude užitečný, ale **bez jakýchkoli záruk**. Dále bez záruk **obchodovatelnosti** nebo **vhodnosti pro konkrétní účel**. Podrobnosti naleznete v GNU Library General Public License.

Spolu s tímto dokumentem byste měli obdržet kopii GNU Library General Public License (jako soubor LICENSE). Pokud nikoli, tak si o ni napište na adresu

Free Software Foundation, Inc., 676 Mass Ave, Cambridge, MA 02139, USA.

## On-line dostupnost

Yetiho typografický bestiář by měl být dostupný na:

PostScript:

<http://trific.ath.cx/Ftp/tex/bestiary.ps.gz>

<http://trific.ath.cx/Ftp/tex/bestiary-booklet.ps.gz>

Zdrojový kód:

<http://trific.ath.cx/Ftp/tex/bestiary.tar.gz>

Informace:

<http://trific.ath.cx/tex,mf/bestiary/>

Nejsem schopen plně garantovat stabilitu uvedených URI, je proto bezpečnější použít oblíbený vyhledávač.

## Poděkování

Chtěl bych poděkovat Josefu Tkadlecovi za vyjasnění některých problémů s mezerami a řadu dalších cenných připomínek, Petru Olšákovi a Hansu Ginzalovi za další připomínky a nápady a mým spolužákům a zároveň prvním čtenářům, za jejich typografické chyby, bez nichž by tento text nikdy nevznikl.

Zároveň se omlouvám všem, z jejichž textů jsem nestydatě opsal příklady, když mne zrovna nenapadl žádný lepší – zejména Donaldu Knuthovi a Petru Olšákovi.