

UNIVERZITA PALACKÉHO V OLOMOUCI
PŘÍRODOVĚDECKÁ FAKULTA
KATEDRA MATEMATICKÉ ANALÝZY A APLIKACÍ MATEMATIKY

BAKALÁŘSKÁ PRÁCE

Sazba matematických formulí L^AT_EXem



Vedoucí bakalářské práce:
RNDr. Miloslav Závodný
Rok odevzdání: 2014

Vypracoval:
Barbora Wyrwolová
ME, III. ročník

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně pod vedením pana RNDr. Miloslava Závodného s použitím uvedené literatury.

V Olomouci dne 30. března 2014

Poděkování

Na tomto místě bych chtěla poděkovat především svému vedoucímu bakalářské práce, který měl se mnou dostatek trpělivosti a pomohl mi tuto práci dovést ke zdárnému konci. Dále bych chtěla poděkovat své rodině a přátelům, kteří mě po dobu mého studia nepřetržitě podporovali.

Obsah

Úvod	4
1 T_EX	5
1.1 Výhody a nevýhody T _E Xu	6
1.1.1 Výhody T _E Xu	6
1.1.2 Nevýhody T _E Xu	7
2 L^AT_EX	9
2.1 Výhody a nevýhody L ^A T _E Xu	9
3 Obecně o formulích	10
3.1 V textu a na řádku	10
3.2 Ukázka formulí	17
4 A_MS-math	21
5 Knihovny v L^AT_EXu	28
5.1 Knihovna yhmath	28
5.2 Knihovna MnSymbol	29
5.3 Knihovna plain	29
5.4 Knihovna latexsym	31
5.5 Knihovna braket	31
Závěr	33
Literatura	34

Úvod

Téma mé bakalářské práce je *Sazba matematických formulí L^AT_EXem*. Důvod, proč jsem si toto téma vybrala je ten, že jsem měla možnost se setkat s texty vysázenými programem T_EX a jejich vzhled se mi líbil mnohem více, než kdyby byl zpracován v jiném programu. Dalším důvodem bylo, že tento program dává možnost sázet matematické vzorce snadněji, než kterýkoliv jiný program, na který jsem byla doposud zvyklá. Chtěla jsem, aby výsledek mé práce byl úhledný, a proto jsem se rozhodla naučit se s tímto programem pracovat.

Cílem mé bakalářské práce je především ukázat, jak je výhodné použít tento program právě když potřebujeme psát matematické vzorce.

Práci jsem rozdělila do pěti přehledných kapitol. V první kapitole najdeme stručné seznámení s programem T_EX. Druhá kapitola vysvětluje, co je L^AT_EX. Ve třetí kapitole se budeme zabývat matematickými formulemi a jejich ukázkami. Ve čtvrté kapitole si ukážeme co je to $\mathcal{A}\mathcal{M}\mathcal{S}$ -math. Poslední kapitola se týká knihoven, které rozšiřují možnosti matematické sazby.

Každý student nějakého matematického oboru se určitě setká nebo již setkal se situací, kdy bude muset psát matematické formule, a to nám T_EX umožňuje jednoduchou a mnohdy i zábavnou formou. Mám za to, že bez zvládnutí sazby systémem T_EXem by naše vzdělání, jakožto absolventů matematického oboru, bylo neúplné.

1. T_EX



Program T_EX byl vytvořen Donaldem Ervinem Knuthem, který byl profesorem Stanfordovy Univerzity. Tento program vytvořil proto, že nebyl spokojen s tím, jak školní nakladatelství vysázelo jeho text. V textu bylo mnoho chyb, především v matematických vzorcích.

Knuth tedy vytvořil program T_EX, který je výhodný pro tvorbu pěkných knih a obzvláště knih, které obsahují spoustu matematiky. T_EX je volně šiřitelný program a nejčastěji je používán matematiky, fyziky a informatiky. Své uplatnění najde i v humanitních oborech (např. při sázení znaků různých abeced), i když podstatný je především bezchybný vzhled výsledku.

T_EX je nezávislý na operačním systému a výstupním zařízení. To znamená, že na všech výpočetních systémech s libovolným operačním systémem pracuje stejně, přitom je jedno, zdali bude výsledek tisknut na laserové tiskárně nebo půjde přes osvitovou jednotku na film.

V T_EXu autor přímo do svého textu zadává různé příkazy, které jasně určují, co by se s textem mělo při sazbě stát. Značky mají formu zpětného lomítka bezprostředně následovaného posloupností znaků, které mají význam písmena a končících mezerou (další mezery už jsou v tomhle programu ignorovány) nebo jedním znakem, který písmeno není. Pomocí tzv. kategorie znaku lze definovat co je a co není písmeno. Značky dělíme na logické (ty nám usnadňují celkový zápis textu) a na optické (pomocí nich ovlivňujeme vzhled sazby). Textový soubor obsahující tyto T_EXovské značky se nazývá *dokument*.

Prvně byl T_EX napsán pouze pro sazbu anglických textů, nyní ho ale již lze přizpůsobit k sazbě v jakémkoliv jazyce. V textu je možné používat i jiné než latinkové abecedy, lze do nich vkládat akcenty, sázet zprava doleva, jak je tomu například v arabštině nebo dokonce do sloupců, jak je tomu v japonštině. Tohle práce je sázena písmem Computer Modern, lze však použít jakékoliv písmo.

Do dokumentu psaného v $\text{\TeX}$ u lze rovněž vkládat obrázky a postskriptové konstrukce.

Při spuštění $\text{\TeX}$ u je nutné, aby program nejprve načetl *formát*, což je knihovna značek v binární podobě, sestavená z tzv. primitivů. $\text{\TeX}$ obsahuje pouze ve svém jádru kolem 300 primitivních značek, s těmi by se ale nepracovalo dobře, jsou vhodné pouze pro zkušené programátory, proto Knuthův koncept $\text{\TeX}$ u vyžaduje formát. Knuth naprogramoval první formát, který nese název *plain* a zakázal ho dále upravovat. To znamená, že jakýkoliv dokument psaný v plainu musí i za 500 let vypadat stejně. $\text{\TeX}$ spuštěný s formátem plain se nazývá Plain $\text{\TeX}$.

Instalaci $\text{\TeX}$ u je možné najít na internetu, stejně tak i spoustu návodů na jeho použití i pomocné studijní literatury. Veškerá potřebná dokumentace je navíc součástí $\text{\TeX}$ u.

Program $\text{\TeX}$ potřebuje ke svému fungování spoustu dalších pomocných programů, které si volá na pomoc, když to potřebuje. Systém $\text{\TeX}$ tedy není pouze jeden program. Tento systém je tedy šířen především pomocí dvou distribucí. Tyto distribuce jsou $\text{\TeX}$ Live a Mik $\text{\TeX}$, které se liší především v tom, jak se instalují. Rozdíly jsou také v tom, jak pracují s fonty. Mik $\text{\TeX}$ vyžaduje, aby měl autor online přístup k internetu. Je to nutné kvůli tomu, že jednotlivé komponenty se doinstalovávají až podle aktuální potřeby uživatele. $\text{\TeX}$ Live se také instaluje z internetu, ale uživatel si přímo při instalaci volí, jaké komponenty chce instalovat a pokud předem neví, které bude používat, je možné zvolit úplnou instalaci.

1.1. Výhody a nevýhody $\text{\TeX}$ u

1.1.1. Výhody $\text{\TeX}$ u

Začneme výhodami $\text{\TeX}$ u:

- Uživatel, který se rozhodne používat $\text{\TeX}$, nemusí být profesionál v oblasti programování a typografické sazby. Má možnost sázet text pomocí jednoduchých a srozumitelných příkazů, které jsou popsány v již zmiňovaných příručkách.

- Pro přípravu zdrojového dokumentu lze využít jakýkoliv textový editor.
- Systém programů je stabilní. Téměř se nezměnil v podstatě od roku 1986.
- Velkou výhodou při sázení matematických vzorců je možnost pracovat s formulemi stejně jako s prostým textem (např. kopírovat bloky).
- Lze snadno tvořit tabulky, vkládat obrázky, členit text na sekce, psát poznámky pod čarou atd.
- Je automatizována tvorba obsahu, rejstříku, seznamu tabulek a obrázků.
- Snadno se lze odkazovat na číslované objekty a stránky nebo na seznam užitých literatury. Křížové reference se automaticky aktualizují.
- Je zaručena stabilita textu. (Na rozdíl od ostatních programů, jako je například Microsoft Office Word, se nemůže stát, aby se při finálních úpravách nebo při tisku změnila celková struktura a úprava textu.)
- $\text{\TeX}$ je oblíben především hlavně díky pohodlné sazbě matematiky, má však pro svou kvalitní sazbu uplatnění i v humanitních vědách (Jednou z prvních knih vysázených Knuthem v $\text{\TeX}$ u byla Bible – podobně jako u Gutenberga).
- Výsledná sazba je v té nejvyšší kvalitě.
- Přes všechny uvedené výhody je $\text{\TeX}$ stále dostupný zdarma.

1.1.2. Nevýhody $\text{\TeX}$ u

$\text{\TeX}$ má ovšem i své nevýhody:

- Aby byl uživatel opravdu schopen kvalitně vysázet text je nutné, aby dobře znal typografická pravidla a ne jen značky $\text{\TeX}$ u. Jakákoliv chyba se totiž výrazně projeví na vzhledu sazby. Navíc, abychom $\text{\TeX}$ u opravdu dobře porozuměli, nestačí nastudovat pouze jednu příručku.

- Velká nevýhodou (alespoň pro začínajícího uživatele) je to, že okamžitě nevidí výsledek své práce. Dokument se nejdříve musí T_EXem zpracovat a až po té je možno si ho prohlédnout na monitoru nebo si ho vytisknout.
- T_EX se automaticky nepřizpůsobuje jazyku, ve kterém chceme psát. Je nutné jazyk nastavit.
- T_EX rovněž nedokáže sám opravit chybu v textu. (Je ale otázkou, do jaké míry je to nevýhoda, své by o automatických opravách říci uživatelé MS Word.)
- Instalace systému T_EX je rozsáhlá, a přestože je její struktura logicky členěna, není snadné v ní najít potřebné knihovny.

2. L^AT_EX

L^AT_EXem nazýváme program T_EX spuštěný spolu s formátem `latex`¹. Formát L^AT_EX byl sestaven Leslie Lamportem. V dnešních dnech je L^AT_EX rozšiřován týmem profesionálů, vedeným Frankem Mittelbachem. Cílem tohoto týmu je sjednotit všechny rozšiřující verze L^AT_EXu, které postupně vznikaly od zavedení L^AT_EXu, dnes označovaného jako L^AT_EX2.09, a dále vylepšovat možnosti tohoto formátu. Aktuální verze formátu se jmenuje L^AT_EX 2_ε, cílem je zmíněná vylepšená verze s pracovním názvem L^AT_EX3. Tato práce se tedy věnuje verzi L^AT_EX 2_ε. Slovo La-TeX je vyslovováno jako „la-tech“. Stručně řečeno, L^AT_EX je typografický značkový program, tudíž potřebuje informace autora o vzhledu sazby. Tyto informace jsou do textu vkládány do programu pomocí značek, nazývaných v L^AT_EXu také příkazy. Autor při práci s L^AT_EXem přímo nevidí, jak bude text po formátování vypadat. Výslednou verzi lze po zpracování prohlédnout na grafickém terminálu nebo vytisknout. Posléze se dá tato verze zpětně upravit nebo přepracovat. Text v L^AT_EX je rozumné napsat tak, aby měl přehlednou logickou strukturu textu.

2.1. Výhody a nevýhody L^AT_EXu

- Při sazbě pomocí L^AT_EXu využívá autor předem definované vzhledy dokumentů – třídy (`\documentclass`). Je k dispozici několik tříd, které jsou vyrobeny profesionály. Díky nim mají dokumenty běžných uživatelů profesionální vzhled.
- Je uživatelský mnohem pohodlnější než plain, nepožaduje od uživatele znalost programování. Uživatel má k dispozici srozumitelné příkazy pro členění textu, může používat automaticky aktualizované křížové reference atd.
- Uživatel má k dispozici balíky `maker` (knihovny), značně rozšiřující možnosti ovládání výsledné sazby. Nevýhodou toho ovšem je, že orientovat se ve všech těchto knihovnách lze jen obtížně a že většinu těchto knihoven běžný uživatel nikdy nepoužije.

¹Nebudeme dělat rozdíl mezi formátem `latex` a programem T_EX spuštěným spolu s ním.

3. Obecně o formulích

Matematický symbol je jakýkoliv znak, který je používán v matematice. Může to být znak pro označení operace s množinami, jejich prvky, čísla či jinými objekty, znak pro množinu, prostor, proměnnou a mnoho dalších matematických objektů. Matematický symbol jako termín vznikl z překladu z angličtiny a i když je hodně používaný, my dáme přednost názvu značka – matematické znaky se sází prostřednictvím značek.

Právě sazba matematiky je nejsilnější stránkou $\text{T}_{\text{E}}\text{X}$ u. Téměř v každé příručce o $\text{T}_{\text{E}}\text{X}$ u najdeme seznam matematických značek, pomocí nichž bychom měli být schopni napsat odborný text, obsahující matematiku. Ne jinak je tomu v případě $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Vědecké texty zabývající se matematikou často vyžadují použití i dalších znaků než jen těch, zařazených do jádra $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u. Jako vhodný prostředek si v tom případě pravděpodobně zvolíme $\mathcal{A}\mathcal{M}\mathcal{S}\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ nebo jiný balíček maker zaměřený na sazbu matematiky.

3.1. V textu a na řádku

Pro sazbu matematiky používá $\text{T}_{\text{E}}\text{X}$ zvláštní mód – matematický. Tento mód se dělí na matematický textový a matematický vysazený.

Matematický textový mód sází matematiku v rámci textového odstavce. Zde uzavíráme matematiku mezi `\( a \)` nebo mezi znaky `$`. Můžeme také použít prostředí `math`. Všechny tyto možnosti jsou naprosto rovnocenné. Tento mód je volán interním příkazem `\textstyle`, můžeme ho ale uvést sami a přepnout se tak do matematického textového módu.

Exponenty (horní indexy) a dolní indexy indikujeme pomocí značek `^` a `_`. Při prvním výskytu značky indexu přejde $\text{T}_{\text{E}}\text{X}$ do `\scriptstyle`, je již v něm, pak do `\scriptscriptstyle`, určeného pro indexy druhé úrovně. Indexy případných dalších úrovní zůstávají v `\scriptscriptstyle`.

Umístění indexů řeší $\text{T}_{\text{E}}\text{X}$. Nejsme-li se tím, jak $\text{T}_{\text{E}}\text{X}$ indexy usadil spokojeni, můžeme před první značku indikující indexy napsat `\limits` nebo `\nolimits`. První značka umisťuje dolní index pod symbol a horní nad symbol, druhá umis-

tuje dolní index u symbolu vpravo dole a horní index vpravo nahoře.

`\prod_{i=1}^{\infty} a_i` ... $\prod_{i=1}^{\infty} a_i$

`\prod_{i=1}\limits^{\infty} a_i` ... $\prod_{i=1}^{\infty} a_i$

`\displaystyle\prod_{i=1}^{\infty} a_i` ... $\prod_{i=1}^{\infty} a_i$

`\displaystyle\prod\nolimits_{i=1}^{\infty} a_i` ... $\prod_{i=1}^{\infty} a_i$

Matematický vysazený mód sází matematiku na samostatný řádek. Knuth pro přepnutí do tohoto módu a posléze jeho ukončení zvolil `$$...$$`, neboť sazba matematiky byla v Americe drahou záležitostí. Tyto přepínače nastaví sazbu ve vysazeném módu na americký standard – centrování na samostatném řádku odděleném od textu. V evropské sazbě se používá i další způsob, zarovnání formulí doleva s odsazením `\mathindent` od levého okraje. Tento způsob se dá nastavit přepínačem `fleqn` uvedeným v nepovinném parametru příkazu `\documentclass`, ten ale při použití `$$...$$` nefunguje. Pro tento případ existují přepínače `[ a \]` nebo prostředí `displaymath`. Tento mód je volán interním příkazem `\displaystyle`, příkaz můžeme ale uvést i sami, přepneme se tak do matematického vysazeného módu.

Vysazené formule můžeme opatřit značkou vpravo pomocí `\eqno` (například `\eqno{(*)}`), anebo značkou vlevo pomocí `\leqno`, značky je nutno uvést až po napsání formule, sází se v matematickém módu.

Mnoho prostředí pro sazbu matematiky přepíná do tohoto módu sazbu automaticky, např. `equation`, `eqnarray` aj. Tato prostředí mohou formule značit číslem (číslovat), pak lze užít příkazy `\label` a `\ref` či `\pageref`.

Sazba v matematickém módu se liší od sazby v textovém režimu:

1. Mezery či ukončení řádku v textu nemají význam, protože sázíme podle speciálních pravidel matematické sazby, kdy je znakům pomocí jejich sazební značky přidělena třída nastavující mezery kolem znaku. Mezery ovšem může autor zadat speciálními příkazy, viz níže.

2. V mnoha prostředích nejsou povoleny prázdné řádky, vzorec musí být napsán v jednom odstavci (tím se však uživatel nemusí příliš zabývat, chybu mu $\text{\TeX}$ oznámí).

3. Kterékoliv napsané písmeno je považované za jméno proměnné a je vysázeno matematickou kurzívou. Jestliže potřebujeme do matematického textu vložit text musíme jej zadat pomocí boxu, např. `\mbox{...}`, `\textrm{...}` apod. Obsahuje-li formule fyzikální veličiny, jejichž indexy značí materiální objekt (resistor, planetu, apod), je nutné sázet indexy vzpřímeně – `\mathrm{...}`.

Všimneme si nyní některých náležitostí matematické sazby.

Tvorba skupin v matematickém režimu

Jestliže je za příkazem s parametrem uvedeno více znaků, je za parametr považován pouze první znak, pokud není parametrem *skupina* znaků – tou je míněna posloupnost znaků mezi složenými závorkami.

Objekty matematických vzorců

V matematických vzorcích se mohou vyskytovat znaky, pro které máme speciální značky:

- Malá písmena řecké abecedy sázíme příkazy typu `\alpha`, `\beta`, ... (α , β , ...) a velká písmena `\Gamma`, `\Delta`, ... (Γ , Δ , ...).
- Chceme-li vysázet odmocninu, použijeme příkaz `\sqrt`, n -tá odmocnina se zadá pomocí volitelného parametru `\sqrt[n]`, např. $\sqrt[4]{1+x}$. Velikost odmocniny L^AT_EX volí sám. K dispozici máme také značku `\surd`, $\sqrt{}$.

Umístění odmocnitele lze změnit příkazy `\leftroot{posun}` a `\uproot{posun}` (z knihovny `amsmath`), kde *posun* je číslo v malých vnitřních jednotkách, udávající jeho velikost. Záporná čísla představují posun v opačném směru, směr posunu je patrný z názvu příkazu.

$$\text{\texttt{\textbackslash sqrt[\textbackslash beta]\{f(x)\}}} \quad \sqrt[\beta]{f(x)}$$

$$\text{\texttt{\textbackslash sqrt[\leftroot{-1}\uproot{3}\textbackslash beta]\{f(x)\}}} \quad \sqrt[\beta]{f(x)}$$

- Potřebujeme-li mít horizontální linku nad nebo pod nějakou částí matematického výrazu, užijeme příkazy `\overline` a nebo `\underline`.

- Když potřebujeme vysázet vodorovnou složenou závorku nad nebo pod matematickým výrazem, užijeme příkazy `\overbrace` a `\underbrace`.

`\underbrace{a\times a\times\dots\times a}_{n\text{trm{-krát}}}`

$$\underbrace{a \times a \times \dots \times a}_{n\text{-krát}}$$

- Vektory můžeme v matematice označit pomocí malé šipky nad proměnnou. Používáme příkaz `\vec`. Podle české normy by ale vektory měly být sázeny bez šipky tučným skloněným písmem, tj. k tomu určeným fontem `bssb10` (ten není standardní) nebo `cmvee10` (ten obsahuje pouze malou abecedu). Skoro stejnou službu ale prokáže i standardní poskriptový font `phvbo8z` (až na vektor a , který by měl být správně ***a***).

`\font\vek=phvbo8z {\vek F a u v p} ... F a u v p`

- Funkce v matematice se nesází kurzívou, ale normálním typem písma. Slouží k tomu kromě jiných například příkazy `\cos`, `\lg`, `\sup`, ... Uživatel si chybějící funkce může snadno dodefinovat, např.

`\newcommand{\tg}{\mathop{\mathrm{tg}}}\tg 45^\circ ... \tg 45^\circ`

- Zlomek můžeme vysázet příkazem `\frac{...}{...}`. Jednoduché zlomky lze zapsat pomocí `/`, např. $1/3$.
- Binomické koeficienty a podobné struktury lze sázet příkazem `\binom`, např. `\binom 52` dá $\binom{5}{2}$. Lze použít i `{... \choose ...}`, `{... \atop ...}` (který nevysází závorky a musíme je přidat) nebo příkaz, který vysází závorky v něm uvedené `{... \atopwithdelims() ...}`.
- Značka integrálu je sázená pomocí příkazu `\int`, značka sumy příkazem `\sum` atd. Meze se zadávají stejně jako indexy.
- Pro závorky a jiné ohraničující značky má $\text{\TeX}$ spousty různých symbolů. Kulaté a hranaté závorky se zadávají odpovídajícími znaky, složené závorky pomocí `\{ a \}`. $\text{\TeX}$ si velikost závorek nenastaví automaticky, tudíž je třeba vybrat jejich velikost příkazy `\big`, `\Big`, `\bigg`, `\Bigg` a jejich levých a pra-

vých analogií, např. `\bigl`, `\Bigr`. Existují i verze pro sazbu mezi znaky, např. `\bigm`.

Věta, že $\text{\TeX}$ si velikost závorek nenastaví automaticky, není až tak pravdivá. Existují párové značky (na řádku se musí vyskytovat obě dvě) `\left` a `\right`. Ty nastaví výšku a hloubku za nimi uvedených omezovačů (závorek) na výšku či hloubku (podle toho, která je větší) symetricky kolem účaří. Např. `\left( \sqrt{\frac{a+b}{2}} \right)^2` dá $\left(\sqrt{\frac{a+b}{2}}\right)^2$.

Potřebujeme-li jen jednu závorku, lze druhou v párové značce nahradit tečkou, např. `\left. \dots \right]`.

- V matematických vzorcích se často užívají tři tečky (tzv. výpustky), ty zadáváme pomocí příkazů `\ldots`, `\cdots`. První příkaz tečky vkládá na základní čáru a druhý sází tečky doprostřed výšky znaků. Lze použít i `\dots`, jak budou ale tyto tečky sázeny ovlivňují použité knihovny (např. `amsmath`) a výsledek se může případ od případu lišit (po , dole, po + uprostřed).

Mezery v matematice

$\text{\TeX}$ sází mezery kolem symbolů automaticky podle jejich třídy. Nejsme-li s tím spokojeni, můžeme je změnit standardními značkami:

Dvojitý (double) quad	<code>\qqquad</code>	
Quad	<code>\quad</code>	
Mezera (space)	<code>\quad</code>	
Velká (thick) mezera	<code>\; \;</code>	
Střední (medium) mezera	<code>\></code>	
Malá (thin) mezera	<code>\,</code>	
Záporná malá mezera	<code>\!</code>	

Samozřejmě lze použít `\hspace` a `\kern`, kde se velikost mezery nastavuje zadanou dimenzí, např. `\hspace{5pt}` a `\kern-1mm`.

Sazba na více řádků ve vysazeném módu

Chceme-li vysázet rovnice nebo dlouhé vzorce, budeme k tomu potřebovat konstrukci, umožňující rozdělit řádky. To přímo nelze, bude nutné použít něco, co se chová podobně jako vertikální box vložený do boxu horizontálního. Takto pracuje např. prostředí `array`, které je analogií prostředí `tabular` pro sazbu tabulek v textovém módu – sloupce jsou nastaveny do `\textstyle`. Prostředí použijeme v matematickém módu a řádky ukončujeme příkazem `\\`. Víceřádkové vzorce (i soustavy rovnic) můžeme sázet také v prostředí `eqnarray` číslující každý řádek (generování čísla řádku lze zrušit uvedením `\nonumber` na tomto řádku) nebo `eqnarray*` (to nečísluje).

Prostředí `eqnarray` je de facto `array` se třemi zamlčenými sloupci `{rcl}`, přitom první a třetí jsou nastaveny do `\displaystyle`, prostřední do `\textstyle`.

K sazbě a číslování jediného řádku slouží prostředí `equation`, které má i nečíslující hvězdičkovou variantu `equation*` (rozhodneme-li se později nečíslovat, nemusíme měnit celé prostředí).

Sázíme-li dlouhé formule, musíme jejich rozdělení na řádky vyřešit sami, $\text{\LaTeX}$ to za nás automaticky neprovede. Autor obvykle nejlépe ví, kde se formule má rozdělit.

Další prostředí pro sazbu formulí přináší $\text{\LaTeX}$ ovské knihovny Americké matematické společnosti $\mathcal{A}\mathcal{M}\mathcal{S}\text{-}\text{\LaTeX}$, konkrétně balíček `amsmath`.

Písmo při sázení matematiky

$\text{\TeX}$ si velikost písma vybírá automaticky sám podle kontextu. Třeba indexy jsou automaticky psány menším písmem než text – první úroveň o 3pt druhá a další o 5pt. Velikost písma, stejně jako rodinu, tučnost a tvar si můžeme nastavit sami. V matematickém módu příkazy

`\mathrm{znaky}` `\mathrm{znaky}` `\mathrm{znaky}`

`\mathnormal{znaky}` `\mathnormal{znaky}` `\mathnormal{znaky}`

`\mathcal{znaky}` `\mathcal{znaky}` `\mathcal{znaky}`

`\mathbf{znaky}` `\mathbf{znaky}` `\mathbf{znaky}`

`\mathsf{znaky}` \$... znaky

`\mathtt{znaky}` \$... znaky

`\mathit{znaky}` \$... *znaky*

Velikost znaků se přizpůsobí stylu:

`\mathcal{A}^{\mathit{B}\mathtt{a}}` \$... $\mathcal{A}^{B^a}$

Je možné použít i textové přepínače, umístění znaků je ale trochu odlišné:

`\textsf{A}^{\textit{B}\texttt{a}}` \$... A^{B^a}

Sazba teorémů

L^AT_EX pro sázení definic, vět a podobných konstrukcí (v řeči L^AT_EXu teorémů) užívá příkaz

`\newtheorem{jméno}[čítač]{text}[oddíl]`.

Argument *jméno* identifikuje danou konstrukci. Argument *text* konkretizuje aktuální jméno naší konstrukce, která bude vysazena. Argumenty, které se nacházejí v hranatých závorkách jsou nepovinné a oba nějakým způsobem ovlivňují číslování dané konstrukce. Argumentem *čítač* se lze odvolat na *čítač* v naší soustavě teorémů – čítače mají stejná jména jako je *jméno* teorému. Pokud se v deklaraci teorému odvoláme na jiný čítač, teorém použije tento čítač (i s případným oddílem) a svůj nebude generovat. Argument *oddíl* je čítač, který bude předcházet číslu teorému, většinou se jedná o číslo nějakého oddílu – sekce, kapitoly, apod.

`\newtheorem{veta}{Věta}[section]`

`\newtheorem{defi}[veta]{Definice}`

Tučné symboly

Je poměrně složité získat v L^AT_EXu tučné symboly. Příkaz `\mathbf` nám sice zajistí tučnost písma, ale pouze pro písmo vzpřímené, ale v matematice většinou znaky píšeme kurzívou. Příkaz `\boldmath` umí pracovat i s matematickými znaky, ale nastavit ho můžeme pouze vně matematického režimu. Nejvýhodnější pro nás je, když uijeme balíček `amssymb`, který je součástí $\mathcal{M}$ S_ET_EXu. Ten nám umožní pohodlnou práci s tučnými matematickými symboly. V tomto balíčku pak

najdeme příkaz `\boldsymbol` a `\pmb`.

3.2. Ukázka formulí

`$x^5+10x+5 = \boldsymbol{0}` `$`

$$x^5 + 10x + 5 = \mathbf{0}$$

`\begin{eqnarray}`

`12x + 2y &= & 45 + 6z \\`

`8z - 4y &= & 3 \\`

`y &= & 1/4`

`\end{eqnarray}`

$$12x + 2y = 45 + 6z \tag{1}$$

$$8z - 4y = 3 \tag{2}$$

$$y = 1/4 \tag{3}$$

`$f(y)=\sqrt[6]{y^3+\frac{\sqrt{7}}{3}}$` $f(y) = \sqrt[6]{y^3 + \frac{\sqrt{7}}{3}}$

`$\frac{z^3}{z+9} \cdot \frac{z+1}{6}$` $\frac{z^3}{z+9} \cdot \frac{z+1}{6}$

`$\displaystyle\lim_{n\rightarrow\infty}\left(1+\frac{1}{n}\right)^n=\mathrm{e}$`

$$\lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n = e$$

`$$`

`f(x)=\left\{\begin{array}{l} \quad \text{pro } x < 0 \\ \quad \text{pro } x = 0 \\ \quad \text{pro } x > 0 \end{array}\right.`

`-1 & x < 0 \\`

`0 & x = 0 \\`

`1 & x > 0`

`\end{array}`

`\right.`

`$$`

$$f(x) = \begin{cases} -1 & \text{pro } x < 0 \\ 0 & \text{pro } x = 0 \\ 1 & \text{pro } x > 0 \end{cases}$$

```
\[
\lim_{n \to \infty}
\sum_{x=1}^n \frac{1}{x^4}
= \frac{\pi^2}{6}
\]
```

$$\lim_{n \rightarrow \infty} \sum_{x=1}^n \frac{1}{x^4} = \frac{\pi^2}{6}$$

```
\begin{displaymath}
\mathbf{Y} =
\left( \begin{array}{ccc}
y_{11} & y_{12} & \ldots \\
y_{21} & y_{22} & \ldots \\
\vdots & \vdots & \ddots
\end{array} \right)
\end{displaymath}
```

$$\mathbf{Y} = \begin{pmatrix} y_{11} & y_{12} & \cdots \\ y_{21} & y_{22} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

```
\begin{equation} \quad \quad \quad \label{moje}
F_{\{X_{i_1},\ldots,X_{i_k}\}}(x_1,\ldots,x_k) =
\prod_{j=1}^k F_{\{X_{i_j}\}}(x_j), \quad \quad \quad \text{forall}
(x_1,\ldots,x_k)\in R^k,
\end{equation}
```

$$F_{(X_{i_1},\dots,X_{i_k})}(x_1,\dots,x_k)=\prod_{j=1}^k F_{X_{i_j}}(x_j), \quad \forall (x_1,\dots,x_k)\in R^k, \tag{4}$$

```

\begin{equation*}
\forall y \in \mathbf{R}:
\quad y^2 \geq 0
\end{equation*}

```

$$\forall y \in \mathbf{R} : \quad y^2 \geq 0$$

```

\begin{displaymath}
y_1, \ldots, y_n, \quad y_1, \ldots, y_n, \quad y_1 + \cdots + y_n, \quad y_1 + \cdots + y_n
\end{displaymath}

```

$$y_1, \dots, y_n, \quad y_1, \dots, y_n, \quad y_1 + \cdots + y_n, \quad y_1 + \cdots + y_n$$

```

\begin{displaymath}
\mathop{\mathrm{corr}}(X,Y)=
\frac{\displaystyle
\sum_{i=1}^n(x_i-\overline{x})
(y_i-\overline{y})}
{\displaystyle\biggl[
\sum_{i=1}^n(x_i-\overline{x})^2
\sum_{i=1}^n(y_i-\overline{y})^2
\biggr]^{1/2}}
\end{displaymath}

```

$$\mathrm{corr}(X,Y) = \frac{\sum_{i=1}^n (x_i - \overline{x})(y_i - \overline{y})}{\left[\sum_{i=1}^n (x_i - \overline{x})^2 \sum_{i=1}^n (y_i - \overline{y})^2 \right]^{1/2}}$$

```

\mathop{\mathrm{corr}}(X,Y)=
\frac{\displaystyle
\sum_{i=1}^n(x_i-\overline{x})
(y_i-\overline{y})}

```

```
{\displaystyle\biggl[
\sum_{i=1}^n(x_i-\overline{x})^2
\sum_{i=1}^n(y_i-\overline{y})^2
\biggr]^{1/2}}
```

$$\text{corr}(X, Y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\left[\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2 \right]^{1/2}}$$

Pokládám za důležité zmínit se o příkazu `openup` z `plainTEXu`, který lze užít ve víceřádkových formulích ke změně prokladu.

```
$$
\openup2pt\displaylines{
a+b+c+d+e+f+g+h\cr 1+2+3+4+5+6+7+8\hfill \cr
\rlap{\$(1)$}\hfill a+1+b+2+c+3-d-4\hfill \llap{\$(*)$} \cr}
$$
```

$$a + b + c + d + e + f + g + h$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8$$

$$(1) \qquad a + 1 + b + 2 + c + 3 - d - 4 \qquad (*)$$

```
$$
\openup2pt\displaylines{
\quad a+b+c+d+e+f+g+h\hfill \cr 1+2+3+4+5+6+7+8 \quad \cr
\hfill a+1+b+2+c+3-d-4 \quad \cr}
$$
```

$$a + b + c + d + e + f + g + h$$

$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8$$

$$a + 1 + b + 2 + c + 3 - d - 4$$

4. $\mathcal{AMS}$ -math

Pro snazší a pohodlnější sázení matematiky byl vytvořen balíček $\mathcal{AMS}$ -math. Balíček $\mathcal{AMS}$ -math je distribuován spolu s dalšími balíčky, rozšiřujícími jeho možnosti. Tento balíček se připojuje příkazem `\usepackage` v preambuli dokumentu. Má tyto volby:

- `centertag` – Tato volba je přednastavena. Číslované prostředí má číslo umístěné uprostřed výšky boxu (centrované na výšku).
- `tbtags` – Číslované prostředí má číslo umístěné vlevo před prvním řádkem, resp. vpravo za posledním řádkem, je-li v dokumentu nastaveno číslování vlevo, resp. vpravo.
- `sumlimits` – Tato volba je přednastavena. Umísťuje indexy standardním způsobem.
- `nosumlimits` – Vždy umístí index vedle symbolu suma (jako `\nolimits`).
- `intlimits` – Tato volba má stejnou funkci jako `sumlimits`, ale v tomto případě pro integrál.
- `nointlimits` – Tato volba je přednastavena. Umísťuje indexy u integrálu standardním způsobem (jako `\nolimits`).
- `namelimits` – index bude sázen pod operátor (např: `min`, `max`, `det`) pokud se vyskytne v prostředí `displaymath`.
- `nonamelimits` – Protiklad `namelimits` (indexy jako `\nolimits`).
- `leqno` – Volitelný parametr nastavující `tbtags`. Číslování formulí je vlevo.
- `reqno` – Volitelný parametr nastavující `tbtags`. Číslování formulí je vpravo.

Všechny tyto volby lze kombinovat, uvádí se v nepovinném parametru při načítání knihovny, např. `\usepackage[intlimits,tbtags]{amsmath}`.

V balíku $\mathcal{AMS}$ -math existují pro sazbu formulí tato prostředí:

equation	equation*	align	align*
gather	gather*	flalign	flalign*
multline	multline*	alignat	alignat*
split			

Prostředí bez hvězdičky automaticky číslují jednotlivé rovnice. Jestliže nechceme, aby určitá rovnice byla číslována, uvedeme na řádku příkaz `\notag` (před `\\`). Symbol `\\` označuje přechod na další řádek, jednotlivé části rovnice se oddělují znakem `&` tak, jako je tomu i v prostředí `array`.

Prostředí $\mathcal{AMS}$ -mathu:

Multline

Toto prostředí je obdobou prostředí `equation`, s tím rozdílem, že ho používáme, jestliže pro napsání rovnice nestačí jeden řádek. Tato dlouhá formule bude očíslována pouze jednou. Jak už bylo zmíněno, lze potlačit její číslování pomocí příkazu `\notag`. Pro umístění neokrajových řádků vlevo a vpravo užíváme příkazy `\shoveleft` a `\shoveright`. Ukázka:

```
\begin{multline}
1+2+2+5+6 \\
+f-b+2+7+c \\
-3a+4+d
\end{multline}
```

$$1 + 2 + 2 + 5 + 6$$

$$+ f - b + 2 + 7 + c$$

$$- 3a + 4 + d \quad (5)$$

Split

Stejně jak `multline`, tak i prostředí `split` slouží k sázení formulí, které jsou dlouhé a nejdou vysázet na jednom řádku. Zarovnání řádků je v tomto prostředí řešeno pomocí značky `&`, jak je dobrým zvykem. Ukázka:

```

\begin{equation}
\begin{split}
f(x,y) &= {}& x^3+10x-4y^3 \\
&& -4y+25
\end{split}
\end{equation}

```

$$f(x,y) = x^3 + 10x - 4y^3 - 4y + 25 \quad (6)$$

Align

Toto prostředí můžeme např. použít pro sazbu dvou a více formulí na jeden řádek. Můžeme přidat i poznámku pomocí příkazu `\text`. Ukázka:

```

\begin{align}
x &= y & x^2+3 &= y \\
x'' &= y'' & x'+3 &= y'+1 \\
f(x) &= x+2 & \text{Lineární rovnice} & \notag
\end{align}

```

$$x = y \qquad x^2 + 3 = y \quad (7)$$

$$x' = y' \qquad x' + 3 = y' + 1 \quad (8)$$

$$f(x) = x + 2 \qquad \text{Lineární rovnice}$$

Příkazem `\intertext` lze do matematické sazby vložit krátký text. Ukázka:

```

\begin{align}
1 &= \cos^3(y) + \sin^4(y) \\
\intertext{nebo}
\sin^3(y) &= 1 - \cos^7(y)
\end{align}

```

$$1 = \cos^3(y) + \sin^4(y) \quad (9)$$

nebo

$$\sin^3(y) = 1 - \cos^7(y) \quad (10)$$

Matice

Balík $\mathcal{A}\mathcal{M}\mathcal{S}$ -math poskytuje i prostředí pro sazbu matic. Jsou to: pmatrix, bmatrix, Bmatrix, vmatrix a Vmatrix. Všechna tato prostředí vychází ze standardního prostředí $\text{\LaTeX}$ u array. Potřebujeme-li vysázet na řádek tečky přes daný počet sloupců uijeme příkaz `\hdotsfor{císlo}`.

Pro estetický dojem jsme použili balíček color – ten lze samozřejmě použít.

Ukázka:

```
{\color{magenta}
$$ \begin{pmatrix}
a_{11} & \dots & a_{1n} \\
\hdotsfor{3} \\
a_{n1} & \dots & a_{nn}
\end{pmatrix}
$}
```

$$\begin{pmatrix} a_{11} & \dots & a_{1n} \\ \dots & \dots & \dots \\ a_{n1} & \dots & a_{nn} \end{pmatrix}$$

Problémem jsou ve standardním $\text{\LaTeX}$ u vícenásobné meze. Lze použít prostředí array, ale pak musíme měnit velikost použitého fontu i prokladu. Bez těchto změn se obejdeme, použijeme-li prostředí subarray.

```
\begin{subarray}{z}
1. řádek \\
2. řádek \\
... \\
poslední řádek
\end{subarray}
```

Argument zarovnání může nabývat hodnoty c nebo l, chceme-li zarovnat řádky vpravo, musíme použít výplně, např.

```
\[ \sum_{\begin{subarray}{l}
i=0,\ldots,9 \\
\hfill j\in \mathbb{N}
\end{subarray}} a_{i,j} \]
```

$$\sum_{\substack{i=0,\dots,9 \\ j\in\mathbb{N}}} a_{i,j}$$

Podobně se chová příkaz `\substack`, který má syntaxi

```
\substack{1. řádek\\ 2. řádek\\ ... \\ poslední řádek}
```

a jehož jednotlivé řádky budou vycentrovány. Použití příkazu je ukázáno na prvním příkladu, v podstatě jím nahradíme pole `array` v indexu.

```
\[ \prod_{\substack{
i=1,\ldots,k \\
k=1,\ldots,l \\
l=k,\ldots,n
}} A_{i+j+k} \]
```

$$\prod_{\substack{i=1,\dots,k \\ k=1,\dots,l \\ l=k,\dots,n}} A_{i+j+k}$$

Speciální meze

V matematickém textu se někdy vyskytují situace, kdy je třeba umístit indexy před symbol, popřípadě na obě jeho strany. K tomuto účelu slouží značka `\sideset`, jejíž syntaxe je `\sideset{před}{za}\symbol` kde `před` a `za` obsahují indexy, které se mají připojit před a za symbol včetně znaku pro jejich umístění. Tedy

`\[ \sideset{_ \alpha ^ \beta }{ _ \epsilon ^ \delta } \\ \bigoplus ^ \gamma _ \iota \]`

dá

$$\bigoplus_{\alpha}^{\beta} \bigoplus_{\iota}^{\gamma} \bigoplus_{\epsilon}^{\delta}$$

Umístit libovolný znak nad symbol či pod symbol v odpovídající velikosti lze také pomocí značek

`\overset{nad}{\symbol}`
`\underset{pod}{\symbol}`

kde nad a pod jsou znaky, které se mají umístit nad a pod symbol ve velikosti prvního indexu. Konstrukce `\overset{\mathrm{def}}{=}` tudíž vytvoří $\stackrel{\mathrm{def}}{=}$. Stejný výsledek získáme pomocí značky `\stackrel{\mathrm{def}}{=}`, např. `\stackrel{\mathrm{def}}{=}` dá $\stackrel{\mathrm{def}}{=}$.

Integrál

Pro sazbu různých typů integrálů máme k dispozici příkazy `\iint`, `\iiint`, `\iiiiint`, `\oint`. Ukázka:

`\def\D{\mathrm{d}}`
`$$\iint \frac{y^3}{y+4} \backslash, \ D y \ $$`
`$$\iint \frac{y^3}{x+4} \backslash, \ D x \ D y \ $$`
`$$\iiint \frac{x^3+y^3+z^3}{y+4} \backslash, \ D y \ D z \ D x \ $$`
`$$\oint \limits_{\Omega} \frac{y^3}{y+4} \backslash, \ D y \ $$`

$$\int \int \frac{y^3}{y+4} dy$$

$$\iint \frac{y^3}{x+4} dx dy$$

$$\iiint \frac{x^3 + y^3 + z^3}{y + 4} \, dy \, dz \, dx$$

$$\oint_{\Omega} \frac{y^3}{y + 4} \, dy$$

5. Knihovny v L^AT_EXu

5.1. Knihovna yhmath

Sice malá, ale nesmírně užitečná knihovna yhmath umožňuje psaní rozlehlých matematických akcentů, což standardně L^AT_EX 2_ε nenabízí. Její zkrácený obsah je tento:

```
% This is file 'yhmath.sty',
\NeedsTeXFormat{LaTeX2e}[1995/12/01]
\ProvidesPackage{yhmath}
\RequirePackage{amsmath}
\DeclareSymbolFont{largesymbols}{OMX}{yhex}{m}{n}
\DeclareMathAccent{\widetilde}{\mathord}{largesymbols}{"65}
\DeclareMathAccent{\widehat}{\mathord}{largesymbols}{"62}
\DeclareMathAccent{\widetriangle}{\mathord}{largesymbols}{"E6}
\DeclareMathAccent{\wideparen}{\mathord}{largesymbols}{"F3}
\newenvironment{amatrix}{\left\langle\begin{matrix}}
{\end{matrix}\right\rangle}
\def\adots{\mathinner{\mkern2mu\raise\p@\hbox{.}
\mkern2mu\raise4\p@\hbox{.}\mkern1mu
\raise7\p@\vbox{\kern7\p@\hbox{.}}\mkern1mu}}
\edef\@tempa#1#2{%
\def#1{\mathaccent\string"\noexpand\accentclass@#2 }}
\@tempa\ring{017}
\newcommand{\widering}[1]{\overset{\smash{\lower1.333ex\hbox{$%
\displaystyle\ring{#1}}}}{\wideparen{#1}}}
\endinput
%% End of file 'yhmath.sty'.
```

Díky ní můžeme sázet např.:

`\verb|$ \widehat{a+b+c} - \widetilde{a-b-b} =`
`\widetriangle{u-w} + \wideparen{u+w} $`

$$\widehat{a+b+c} - \widehat{a-b-b} = \widehat{u-w} + \widehat{u+w}$$

nebo

`$ \widering{ABCD} \adots ABCD $` $\overset{\circ}{\widehat{ABCD}} \cdots ABCD$

5.2. Knihovna MnSymbol

Nejsme-li spokojeni s matematickými symboly nabízenými L^AT_EXem, resp. $\mathcal{M}\mathcal{S}$ -T_EXem, můžeme použít knihovnu MnSymbol, kterou připojíme příkazem `\usepackage{MnSymbol}` v preambuli dokumentu. Knihovna MnSymbol je nekompatibilní s knihovnami amssymb a amsfonts, automaticky načítá knihovnu amsmath. Knihovna používá fonty MnSymbol, kaligrafické písmo lze přepnout na rodinu cmsy, znaky nikoliv.

Seznam všech dostupných symbolů lze nalézt v dokumentaci `texmf-dist/doc/latex/mnsymbol/MnSymbol.pdf`.

5.3. Knihovna plain

Možnosti sazby matematiky značně rozšiřuje knihovna plain. Zná-li uživatel příkazy plainT_EXu, může je pohodlně použít v prostředí plain. Např.

```
\begin{plain}
$$
\openup2pt \left. \eqalign{
{\cal B}(\alpha,\beta)&=\int_0^1 x^{\alpha-1}(1-x)^{\beta-1}dx\cr
{\cal B}(\alpha,\beta)&={\cal B}(\beta,\alpha) \cr
{\cal B}(\alpha,\beta)&={\Gamma(\alpha)\Gamma(\beta)}\cr
\over {\Gamma(\alpha+\beta)}}\right. \eqno{(3)}
$$
```

`\end{plain}`

$$\left. \begin{aligned} \mathcal{B}(\alpha, \beta) &= \int_0^1 x^{\alpha-1} (1-x)^{\beta-1} dx \\ \mathcal{B}(\alpha, \beta) &= \mathcal{B}(\beta, \alpha) \\ \mathcal{B}(\alpha, \beta) &= \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} \end{aligned} \right\} \quad (3)$$

nebo `\eqalignno{}` a `\leqalignno{}` pro číslované rovnice

`\begin{plain}`

`$$`

`\leqalignno{`

`a_1+a_2+a_3=0& (1) \cr`

`3a_3=21& (2) \cr`

`12a_1+24a_2=36& (3) \cr}`

`$$`

`\end{plain}`

$$(1) \qquad a_1 + a_2 + a_3 = 0$$

$$(2) \qquad 3a_3 = 21$$

$$(3) \qquad 12a_1 + 24a_2 = 36$$

nebo `\matrix{}` a její varianty pro sazbu matic

`\begin{plain}`

`$$M=\pmatrix{`

`\pmatrix{\lambda & 1\cr 1 & \lambda\cr} & \lambda\cr`

`1&\pmatrix{1 & 0 \cr \lambda & 1\cr}`

`\cr}$$`

`\end{plain}`

$$M = \begin{pmatrix} \begin{pmatrix} \lambda & 1 \\ 1 & \lambda \end{pmatrix} & \lambda \\ 1 & \begin{pmatrix} 1 & 0 \\ \lambda & 1 \end{pmatrix} \end{pmatrix}$$

Lze použít všechny matematické konstrukce.

5.4. Knihovna latexsym

Tato knihovna existuje od vzniku verze $\text{\LaTeX} 2_{\epsilon}$. Ve starší verzi $\text{\LaTeX} 2.09$ byly čtyři značky definované v této knihovně součástí jádra $\text{\LaTeX} u$. Vyřazení značky `\Box` užívané k ukončení důkazů vět z jádra nebylo šťastné. Obsah knihovny je tento:

```
\DeclareSymbolFont{lasy}{U}{lasy}{m}{n}
\SetSymbolFont{lasy}{bold}{U}{lasy}{b}{n}
\DeclareMathSymbol\Box{\mathord}{lasy}{"32}
\DeclareMathSymbol\Join{\mathrel}{lasy}{"31}
\DeclareMathSymbol\lhd{\mathbin}{lasy}{"01}
\DeclareMathSymbol\rhd{\mathbin}{lasy}{"03}
```

□ ✕ ◁ ▷

5.5. Knihovna braket

Definuje příkazy pro psaní závorek v matematice, a to

```
\bra{ } \ket{ } \braket{ } \set{ } (malá verze)
\Bra{ } \Ket{ } \Braket{ } \Set{ } (expandovaná verze)
```

Expandovaná verze má omezovače uvedené za příkazy `\left` a `\right`. Např. `\Set{ x \in \mathbf{R}^2 \mid 0 < |x| < 5 }` dá $\{ x \in \mathbf{R}^2 \mid 0 < |x| < 5 \}$, resp.

$$\{ x \in \mathbf{R}^2 \mid 0 < |x| < 5 \}.$$

`\Braket{ \phi | \frac{\partial^2}{\partial t^2} | \psi }` vysází

$$\left\langle \phi \left| \frac{\partial^2}{\partial t^2} \right| \psi \right\rangle$$

První dvě značky pracují takto: `$\bra{a+b}$` a `$\ket{a+b}$` $\langle a + b|$ a $|a + b\rangle$

Závěr

Ve své bakalářské práci jsem se zabývala programem $\text{\LaTeX}$ především v souvislosti se sázením matematiky. Popsala jsem nejen všechny příkazy potřebné k sázení matematiky, ale taktéž i samotný $\text{\TeX}$ a $\text{\LaTeX}$. Mým cílem bylo porozumět jednotlivým příkazům – způsobu jejich zadávání i jejich vlivu na výslednou sazbu.

Hlavním cílem práce bylo ukázat, že práce s $\text{\LaTeX}$ em není až tak složitá, jak se na první pohled může zdát. Po nastudování stručné příručky se s tímto programem dá snadno pracovat. $\text{\LaTeX}$ může využít kdokoliv, kdo píše o matematice a potřebuje ve svém textu různé vzorce a matematické formule.

Tím, že jsem s $\text{\LaTeX}$ em pracovala jsem se utvrdila v přesvědčení, že kdykoliv v budoucnu budu potřebovat psát práci obsahující matematiku, tak jediné v $\text{\LaTeX}$ u. Stejně bych doporučila všem studentů, kteří studují nejen matematiku a fyziku, ale i humanitní vědy. Důkladněji jsem se s $\text{\LaTeX}$ em seznámila až při psaní bakalářské práce, tudíž zpočátku bylo komplikovanější najít správný příkaz pro jistý matematický znak, ale bohatá dokumentace mi pomáhala. Čím déle jsem s $\text{\LaTeX}$ em pracovala, tím rychleji mi šla práce od ruky. Mohu říci, že $\text{\LaTeX}$ mi ušetřil hodiny práce, než kdybych použila jiný program.

Budu ráda, když moje práce bude přínosem i pro ostatní studenty. Díky mnoha ukázkám, obsaženým v mojí práci totiž uvidí, jak snadno a pohodlně se dá napsat text obsahující matematické znaky.

Literatura

- [1] [www. wikipedia.com](http://www.wikipedia.com)
- [2] Rybička, J.: L^AT_EX pro začátečníky (3. vydání). Konvoj, Brno, 2003.
- [3] Závodný, M.: Možnosti pro sazbu matematiky v AmsLaTeX u
- [4] Doob, M. : Jemný úvod do T_EXu
- [5] Oetiker, T.: Ne příliš stručný úvod do systému L^AT_EX