VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

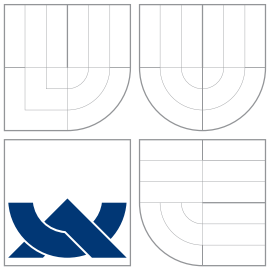
INTERPRET JAZYKA LaTeX ZALOŽENÝ NA KOMBINACI VÍCE METOD SYNTAKTICKÉ ANALÝZY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

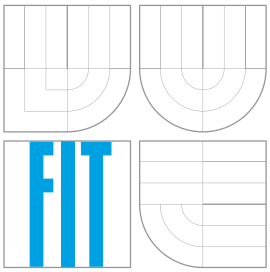
AUTOR PRÁCE
AUTHOR

Bc. PETR LEBEDA

BRNO 2008



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INFORMAČNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INFORMATION SYSTEMS

INTERPRET JAZYKA LaTeX ZALOŽENÝ NA KOMBINACI VÍCE METOD SYNTAKTICKÉ ANALÝZY

LaTeX INTERPRETER BASED ON DIFFERENT TYPES OF SYNTAX ANALYSIS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE — Bc. PETR LEBEDA
AUTHOR

VEDOUCÍ PRÁCE — prof. RNDr. ALEXANDER MEDUNA, CSc.
SUPERVISOR

BRNO 2008

Abstrakt

Tato práce pojednává o možnostech interpretace typografického jazyka LaTeX a popisuje jeho strukturu, funkce a jeho syntaxi. Rovněž rozebírá možnosti interpretace LaTeXu do HTML kódu (HyperText Markup Language) tak, aby byly vytvářeny typograficky přesné publikace, které by bylo možno prohlížet běžným internetovým prohlížečem. Následuje návrh řešení a nastíněné problémy.

Klíčová slova

LaTeX, interpret, syntaktická analýza, HTML, syntaktická analýza shora dolů a zdola nahoru, analýza rekurzivním sestupem

Abstract

The diploma thesis discusses the potential of interpretation of typographical language LaTeX and describes the structure of this language, its functions and their syntax. Also it analyses possibilities of LaTeX interpretation into HTML (HyperText Markup Language), in order to create typographically accurate publications, which could be viewed by common web browser. The solution concept and outlines of possible problems follows.

Keywords

LaTeX, interpretation, syntax analysis, HTML, top-down parsing, bottom-up parsing, recursive descent parsing

Citace

Petr Lebeda: Interpret jazyka LaTeX založený na
kombinaci více metod syntaktické analýzy, diplomová práce, Brno, FIT VUT v Brně, 2008

Interpret jazyka LaTeX založený na kombinaci více metod syntaktické analýzy

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana profesora Meduny. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Petr Lebeda
6. května 2008

Obsah

Kapitola 1

Úvod

Interpretací můžeme rozumět převod informace z jedné její podoby do druhé, aniž by utrpěl obsah informace. Příkladem interpretace může být simultální překlad z jednoho jazyka do druhého. Pokud se ale na intepretaci pohlíží z hlediska výpočetní techniky, jedná se spíše o způsob zpracování vstupních dat a jejich prezentace, případně jejich příprava pro další zpracování.

V případě této diplomové práce se jedná o zpracování vstupních dat, což je zdrojový soubor typografického systému LaTeX. Výstupem by pak měl být soubor, který prezentuje vstupní data ve viditelné formě pro uživatele, avšak zůstává otázkou, jaký formát výstupního souboru zvolit.

Jako interpretaci LaTeXu použiji převod zdrojového souboru psaného v LaTeXu do HTML kódu. Hlavní myšlenkou je, aby i na webových stránkách existovaly typograficky korektní, úhledně psané dokumenty, aniž by musely být ve formátu PDF. Tato problematika je sice již poměrně dlouho řešena, bohužel ale většinou na platformě operačních systémů UNIX. Pro operační systém společnosti Microsoft je teoreticky využitelný projekt doc++, tento je ale stále ve fázi vývoje, více viz [13].

Již rozdíly ve využití těchto dvou programovacích jazyků nutí hledat rozumný kompromis tak, aby bylo možné interpretovat LaTeX do HMTL kódu a přitom byl pokud možno identický s PDF souborem, vytvořeným z téhož LaTeXového zdrojového souboru. To s sebou nese nutná omezení velmi robustního nástroje, jakým LaTeX bezesporu je. Matematické vzorce a obrázky přímo generované LaTeXem nelze snadno převést do HTML. Nezbývá tedy, než z LaTeXu tyto části kódu, kde je daná matematická rovnice, vyčlenit, vytvořit z nich obrázek a tento následně vložit do HTML kódu. Pro konečné odsazení textu a dodržení definovaného formátu se nabízí použití kaskádových stylů CSS (viz [11]).

Následuje lehký úvod do obou programovacích jazyků, jak LaTeXu, tak i HTML a CSS.

1.1 Úvod do HTML

HTML je zkratka pro výraz HyperText Markup Language, což by se dalo přeložit asi jako značkovací jazyk hypertextu. Hypertext umožňuje propojení internetové stránky na jakoukoliv jinou stránku, kdekoliv v celé internetové síti WWW (World Wide Web).

Autorem HTML jazyka je Tim Berners-Lee a Robert Cailliau. Jejich spoluprací vznikl projekt *World Wide Web project* (viz [17]). První veřejně dostupný popis HTML byl uveden v dokumentu *HTML Tags*, viz [18]. Popisuje 22 značek, popisujících prvotní design HTML. Třináct těchto značek přetrvalo ve standartu HTML dodnes. Od první verze HTML, verze

1.0, uvedené v roce 1993, došlo k rozšířením až k aktuální verzi 4.01.

Dokumenty napsané v HTML se ukládají jako textové soubory s příponou *.html*, které jsou poté zobrazeny libovolným prohlížečem (Microsoft Internet Explorer, Mozilla Firefox, Opera a další). Univerzálnost tohoto jazyka by měla zaručovat stejné zobrazení na všech prohlížečích. V minulosti se ale standardy příliš nedodržovaly, což mělo za následek nutnost vytvářet pro každý prohlížeč specificky upravené zdrojové HTML soubory tak, aby jednotlivé stránky byly skutečně zobrazitelné na všech prohlížečích.

Bylo nutné přijmout jednotný standart. Tvorbu standartů má na starosti organizace W3C (World Wide Web Consortium) viz [12].

Značkovací jazyk HTML se skládá z několika klíčových komponent, značek a jejich atributů a dalších znakových entit. Značky, atributy i entity jsou pevně dané, tvůrce internetového dokumentu tak nemůže definovat nové značky ani entity. Text dokumentu může využívat libovolnou standartní znakovou sadu.

Struktura HTML dokumentu je následující:

```
<!DOCTYPE html PUBLIC
"-//W3C/DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml/DTD/xhtml1-transitional.dtd">

<html mlns="http://www.w3.org/1999/xhtml">
<head>
⋮
⋮       hlavička HTML
⋮
</head>
<body>
⋮
⋮       tělo, textová část
⋮
</body>
</html>
```

V hlavičce se zpravidla deklaruje kódování znaků HTML dokumentu. Zpravidla se tak děje elementem

```
<meta http-equiv="content-type" content="text/html; charset=kódování"/>,
```

kde *kódování* je nahrazeno názvem daného kódování. Pro systém Microsoft Windows je výchozí kodování windows-1250. Dále může být v hlavičce definován titulek, zobrazovaný v horní liště prohlížeče.

Tělo zdrojového textu obsahuje text, který má být zobrazen na internetových stránkách, obklopený značkami HTML, které upřesňují způsob zobrazení textu.

Značky se v HTML zapisují podle syntaxe SGML. Každá značka tvoří tzv. HTML element, jehož jméno odpovídá jménu značky a jeho obsah odpovídá obsahu této značky. Element tedy může obsahovat vnořené elementy odpovídající vnořeným značkám. Značky jsou obvykle ve dvojicích, označující uvozující a uzavírací značky. Pro zvýraznění textu

lze použít příkaz `<em>zvýrazněný text</em>`, kde `<em>` je uvozující element a `</em>` je ukončující element.

U nepárových značek je element tvořen pouze touto značkou a nemá žádný obsah. V HTML se nerozlišují malá a velká písmena v názvech značek ani atributů.

Elementy HTML se dále dělí na blokové a řádkové elementy. Řádkové elementy formátují vzhled textu, neovlivňují ale již tok textu. Jejich použití tedy nezpůsobí zalomení řádku. Oproti tomu blokové elementy tok textu ovlivňují, k zalomení řádku dochází na každém začátku i konci blokových elementů. Podle typu elementu a v závislosti na dalších faktorech (např. úprava elementů pomocí CSS) může dojít k odsazení, vynechání volného místa, atd.

Podrobnější popis HTML elementů je např. v [2] anebo v [1]. Pro účely diplomové práce budou elementy HTML srovnány s příkazy LaTeXu v dodatcích B a C.

1.2 Úvod do CSS

Výraz CSS (viz [11]), neboli Cascading Style Sheets (listy kaskádových stylů) úzce souvisí s jazykem HTML a internetovými stránkami jako takovými. Kaskádové styly představují nástroj pro specifikaci vzhledu elementů HTML. Stylování internetových stránek pomocí CSS přináší výhody hlavně v přehlednosti definice vzhledu, její relativně snadné modifikovatelnosti a také to, že jeden kaskádový styl může být sdílen více HTML dokumenty.

První verze CSS, zvaná *CSS Level 1*, vznikla v roce 1996. V té době však podpora CSS v internetových prohlížečích nebyla postačující. Všechny prohlížeče kaskádové styly nezobrazovaly správně, anebo vůbec. Z těchto důvodů nebylo CSS příliš oblíbeno a používáno. Změnu přinesla až nová generace internetových prohlížečů s dobrou podporou CSS.

Aktuální verzí CSS je verze CSS2, která sice ještě není plně podporována prohlížeči, ale i přes tyto nedostatky se běžně v praxi používá. Její nedostatky by měla odstranit chystaná třetí verze CSS3 ([2, strana 57-58]).

Základní jednotkou definice stylu dokumentu je takzvaný *stylový předpis – style sheet*. Je to textový soubor, s příponou *.css*, obsahující posloupnost pravidel definujících vzhled HTML elementů zde uvedených.

Každé pravidlo se skládá ze selektoru a deklarace vlastností:

```
selektor { deklarace vlastností }
```

Selektor určuje množinu HTML elementů, na které se dané pravidlo aplikuje. Množina HTML elementů je definována podle jmen těchto elementů, hodnot jejich atributů a dalších kritérií.

Deklarace vlastností se pak týká vizuálních, či jiných vlastností elementů. Pomocí těchto deklarací je možné měnit barvu textu v jednotlivých odstavcích, či upravit způsob zobrazení nadpisů:

```
h1{ font-size: 22px; font-variant: small-caps; color: red}
```

Pravidlo uvedené výše způsobí, že v HTML dokumentu používajícím daný CSS styl, se budou nadpisy první úrovně zobrazovat o velikosti 22 pixelů kapitálkami a červenou barvou.

Specifikace CSS obsahuje rozsáhlou množinu vlastností, které může element HTML mít. Některé vlastnosti se týkají všech elementů, jiné budou platit jen pro některé z nich. Každá tato vlastnost má dané jméno a obor hodnot, kterých může nabývat.

Navíc každý HTML element může patřit do různých tříd a tím jsou dostupné další možnosti jak nastylovat HTML dokument. Propojení s HTML dokumentem může být

vložením stylového předpisu přímo do HTML dokumentu, nebo se vytvoří soubor s příponou *.css*, který je následně odkazován z hlavičky HTML souboru. ([2, strana 58-61]).

Pro funkce tohoto diplomového projektu bude zřejmě nejlepší druhá možnost, vytvoření samostatného *.css* souboru. Více podrobností o CSS a jeho provázání s HTML viz [2].

1.3 Úvod do LaTeXu

LaTeX, vyslovovaný jako "lejtek" či "lejtech", je program pro sazbu typograficky kvalitních textů. Je jednou z nejrozšířenějších nadstaveb systému TeX, jehož autorem je pan Donald Knuth (více např. [15]). LaTeX je převážně využíván na akademické půdě, hlavně díky jeho možnostem precizně sázet matematické vzorce. Toto ale není jeho jediná schopnost, pomocí něj je možno sázet nepřeberné množství typů publikací, knihami počínaje, přes články, technické zprávy, slajdy a dopisy konče.

Autorem LaTeXu je pan Leslie Lamport (více o celém projektu LaTeX viz [16]). LaTeX je neustále vyvíjen. Celosvětově se stala velmi rozšířenou verze LaTeX 2.09. K této verzi vznikla celá řada doplňků, které vytvořili sami uživatelé. To však vedlo k nekompatibilitě celého systému. Bylo proto rozhodnuto vytvořit projekční tým, jehož konečným cílem je vytvořit LaTeX verze 3. Do té doby, než tento bude hotový, platí standard nazvaný LaTeX 2ε ([3, strana 11]).

Jádro systému tvoří překladač jazyka TeX společně s nadstavbou LaTeX, jehož vstupem je textový soubor, vytvořený libovolným textovým editorem. Tento soubor obsahuje kromě textu, který má být vysázen, také příkazy jazyka LaTeX jak má být tento text vysázen.

Každý dokument určený ke zpracování má následující strukturu:

```
\documentclass[volby]{třída}[datum vytvoření]
⋮
⋮        hlavička
⋮
\begin{document}
⋮
⋮        tělo, textová část
⋮
\end{document}
```

V povinném úvodním příkazu `\documentclass` je parametr třída, definující styl sazby. Standartní třídy jsou `article` - článek, `report` - zpráva, `book` - kniha a `letter` pro dopis.

Příkaz `\documentclass` může mít ještě volitelné parametry, určující další možnosti sazby dokumentu. Je možné například nastavit velikost písma, způsob zobrazení dokumentu (na výšku či na šířku), či rozměry formátu dokumentu (A4, A5, ...).

V hlavičce zdrojového souboru lze také připojovat balíčky dalších příkazů – *packages*. Tím máme možnost rozšířit funkčnost systému například o příkazy zpracující manipulaci s obrázky, nové sady fontů a další. Balíčků je nepřeberné množství, v diplomovém projektu je však nevyužívám, nebudeme je tedy dále rozvádět.

Textová část zdrojového souboru obsahuje již samotný text, který má být vysázen, obklopený příkazy systému LaTeX. Příkazy určují formátování textu, ale upravují také délkové registry, používané pro odsazování různých částí textu. Příkazy mohou být uvedeny

samostatně, např. příkaz `\noindent` způsobující potlačení odsazení prvního řádku odstavce, nebo ve skupinách, např. `\textit{text}`, způsobující vysazení textu v závorkách *kurzivou*.

Další skupinou příkazů jsou takzvaná prostředí. Jedná se o dvojici příkazů `\begin{název}` a `\end{název}`, mezi nimiž je nějaký úsek textu, který je chápán jiným způsobem než okolní text (jiný způsob sazby, vložení obrázku, tabulky a jiné).

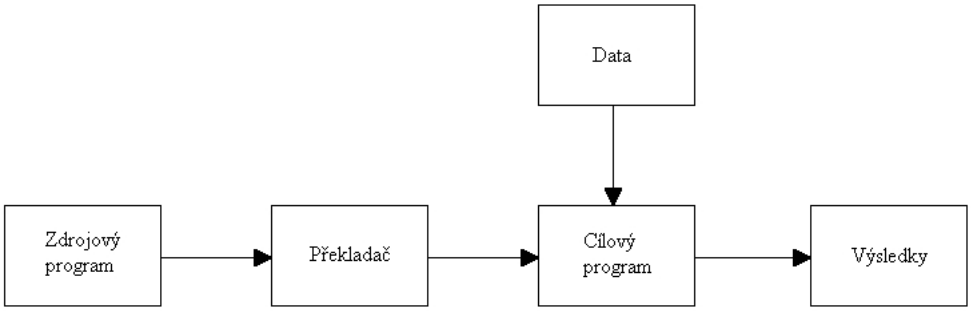
Přehled prostředí a funkcí, se kterými diplomový projekt pracuje, je uveden v dodatku B, včetně jejich stručného popisu.[1]

[1] Při tvorbě této kapitoly jsem čerpal hlavně z knihy Jiřího Rybičky [3], ze stejného zdroje je i zmiňovaný dodatek B

Kapitola 2

Konstrukce interpreteru

Je-li program napsaný v některém vyšším programovacím jazyce, existuje několik možných přístupů k jeho spuštění. Buď je možno program převést do ekvivalentního programu ve strojovém kódu počítače – překladače tohoto typu se označují názvem kompilátory nebo kompilační překladače. Nebo je možno napsat program, který bude interpretovat příkazy zdrojového jazyka tak, jak jsou napsané, a přímo provádět odpovídající akce. Programy realizující druhý přístup se nazývají interprety nebo interpretační překladače. Obrázky 2.1 a 2.2 představují schémata činnosti obou typů překladačů.



Obrázek 2.1: Kompilační překladač

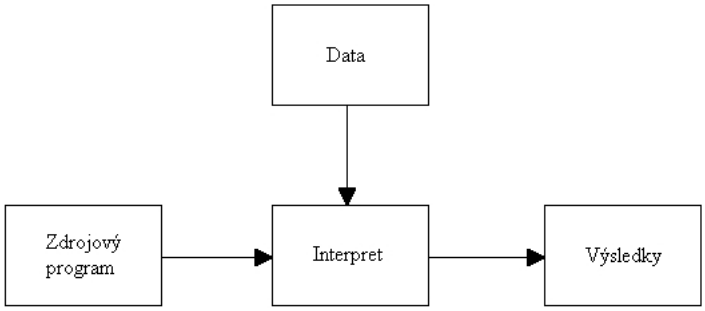
Výhodou kompilace je, že analýza zdrojového programu a jeho překlad se provádějí jen jednou, i když může jít o časově dosti náročný proces. Dále je již spouštěn pouze ekvivalentní program ve strojovém kódu, který je výsledkem překladu.

Kód generovaný kompilačním překladačem nemusí být obecně ekvivalentní se strojovým kódem nějakého konkrétního počítače.

Interpretace je mnohem pomalejší než kompilace, neboť je třeba analyzovat zdrojový příkaz pokaždé, když na něj program narazí. Pro poměr mezi rychlostí interpretovaného a kompilovaného programu se uvádějí hodnoty mezi 10:1 až 100:1, v závislosti na konkrétním jazyce. Interprety bývají také náročné na paměťový prostor, neboť i při běhu programu musí být stále k dispozici celý překladač.

Interprety však mají i své výhody oproti kompilačním překladačům. Při výskytu chyby jsou vždy přesné informace o jejím výskytu a je možno poměrně rychle odhalit její příčinu. Tento přístup je tedy vhodný zvláště při ladění programů. Interprety umožňují modifikaci textu programu i během jeho činnosti, což se využívá často u jazyků jako je Prolog nebo LISP. U jazyků, které nemají blokovou strukturu (např. BASIC, APL), se může změnit

některý příkaz, aniž by se musel znovu překládat zbytek programu. Interprety se dále používají tam, kde se mohou typy objektů dynamicky měnit v průběhu provádění programu – typickým příkladem je jazyk Smalltalk-80. Jejich zpracování je pro kompilační překladače značně obtížné. Interpretační překladače bývají značně strojově nezávislé, protože negenerují strojový kód. Pro přenos na jiný počítač obvykle postačí interpret znovu zkompilovat.



Obrázek 2.2: Interpretační překladač

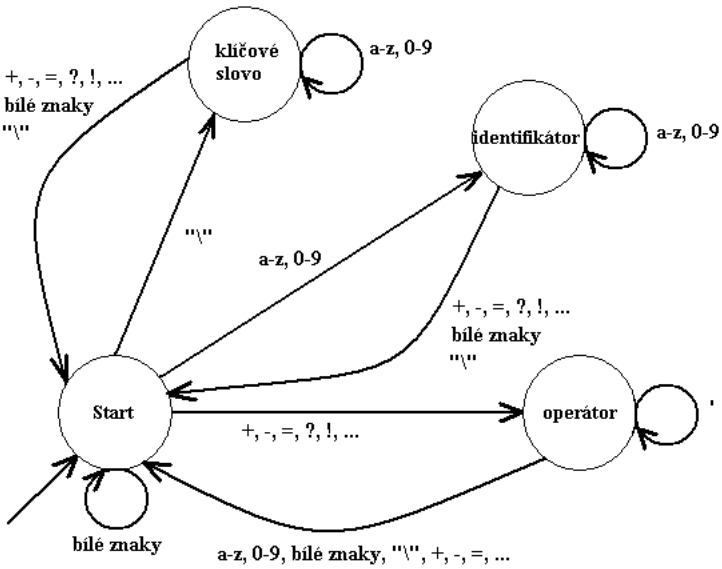
Uvedené dva přístupy jsou však extrémní, mnoho překladačů využívá spíše jejich kombinace. Některé interpretační překladače například nejdříve převedou zdrojový program do nějakého vnitřního tvaru (v nejjednodušším případě alespoň nahradí klíčová slova jejich binárními kódy) a ten potom interpretují. Výsledné řešení je kompromisem mezi časově náročným překladem kompilovaného a pomalým během interpretovaného programu.

Interpreter využitý v této diplomové práci se poněkud liší od výše uvedených příkladů. Nevykonává početní příkazy, ale na základě příkazů LaTeXu, vypíše do výstupního souboru příkazy pro HTML. HTML kód je poté vizuálně intepretován v některém z internetových prohlížečů.

Kapitola 3

Lexikální analýza

Fáze lexikální analýzy (lexical analysis, scanning) čte znaky zdrojového programu a sestavuje je do posloupnosti lexikálních symbolů, v níž každý symbol představuje logicky související posloupnost znaků jako identifikátor nebo operátor obdobný \\. Posloupnost znaků tvořících symbol se nazývá lexém (lexeme).



Obrázek 3.1: Scanner

Po lexikální analýze znaků např. v tomto přiřazovacím příkazu

```
pozice := počátek + rychlost * 60
```

by se vytvořily následující lexikální jednotky:

1. identifikátor pozice
2. symbol přiřazení :=
3. identifikátor počátek
4. operátor +

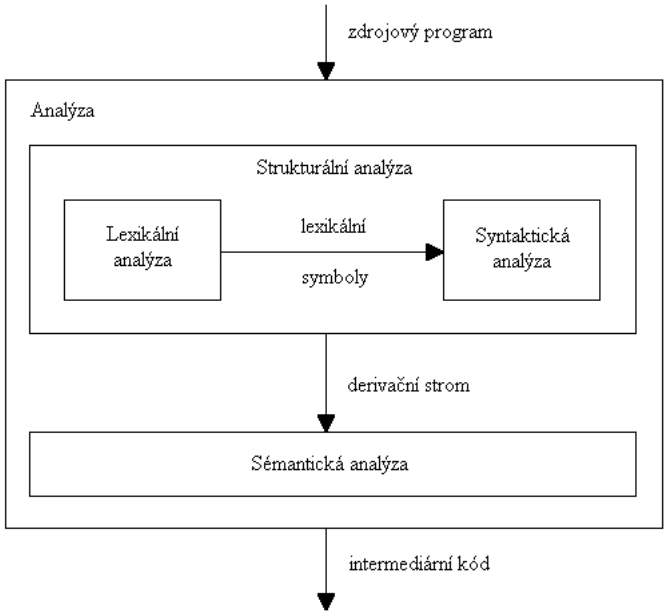
5. identifikátor rychlost
6. operátor *
7. číslo 60

Symboly, které zahrnují celou třídu lexikálních jednotek (identifikátor, číslo, řetězec), jsou reprezentovány obvykle jako dvojice `<druh symbolu, hodnota>`, přičemž druhá část dvojice může být pro některé symboly prázdná. Výstupem lexikálního analyzátoru pro příkaz (1.1) by tedy mohla být posloupnost

```
<id,pozice> <:=> <id,počátek> <+> <id,rychlost> <*> <num,60>
```

Mezery, konce řádků a poznámky oddělující lexikální symboly se obvykle během lexikální analýzy vypouštějí.

Lexikální analyzátor - scanner - použitý v tomto programu by se dal chápat jako konečný automat, který se po přečtení určitého znaku přesune do určitého stavu. Po přečtení dalšího znaku v tomto stavu setrvává, anebo provede uložení načteného lexemu a vrátí se do startovního stavu. Tento proces je ukázán na obrázku scanneru, viz 3.1. Tento obrázek demonstruje konečný automat využitý přímo v tomto interpreteru.

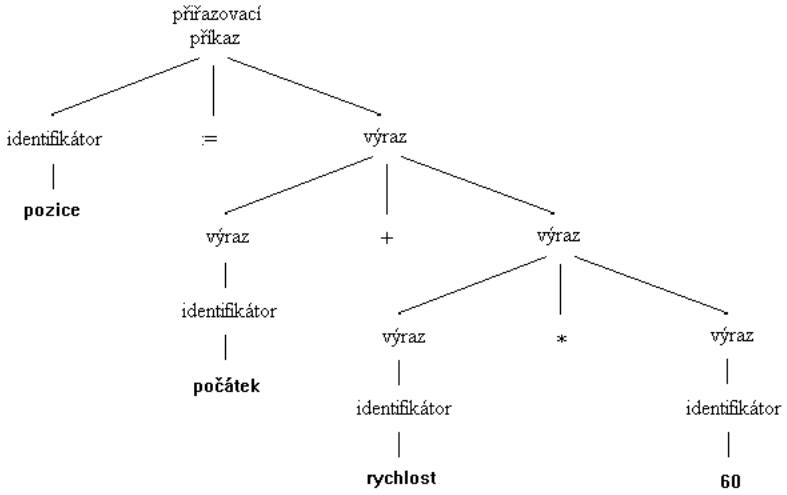


Obrázek 3.2: Struktura analytické části překladače

Kapitola 4

Syntaktická analýza

Syntaktická analýza (parsing, syntax analysis) spočívá v sestavování lexikálních jednotek ze zdrojového programu do gramatických frází, které překladač používá pro syntézu výstupu. Gramatické fráze zdrojového programu se obvykle reprezentují derivačním stromem obdobným stromu na obr. 4.1.



Obrázek 4.1: Derivační strom pro výraz pozice := počátek + rychlost * 60

Ve výrazu počátek+rychlost*60 je fráze rychlost*60 logickou jednotkou, protože podle běžných matematických konvencí pro aritmetické výrazy se násobení provádí před sčítáním. Vzhledem k tomu, že za výrazem počátek+rychlost následuje *, nevytváří tento výraz v situaci na obr. 4.1 frázi.

Hierarchická struktura programu se obvykle vyjadřuje pomocí rekurzivních pravidel, zapsaných ve formě bezkontextové gramatiky. Například pro definici části výrazu můžeme mít následující pravidla:

$$\text{výraz} \; -> \; \text{identifikátor} \tag{4.1}$$
$$\text{výraz} \; -> \; \text{číslo} \tag{4.2}$$
$$\text{výraz} \; -> \; \text{výraz} + \text{výraz} \tag{4.3}$$
$$\text{výraz} \; -> \; \text{výraz} * \text{výraz} \tag{4.4}$$
$$\text{výraz} \; -> \; (\ \text{výraz}\ ) \tag{4.5}$$

Pravidla (4.1) a (4.2) jsou (nerekurzivní) základní pravidla, zatímco (4.3)–(4.5) definují výraz pomocí operátorů aplikovaných na jiné výrazy. Podle pravidla (4.1) jsou tedy `počátek` a `rychlost` výrazy. Podle pravidla (4.2) je `60` výraz, zatímco z pravidla (4.4) můžeme nejprve odvodit, že `rychlost*60` je výraz a konečně z pravidla (4.5) také `počátek + rychlost * 60` je výraz.

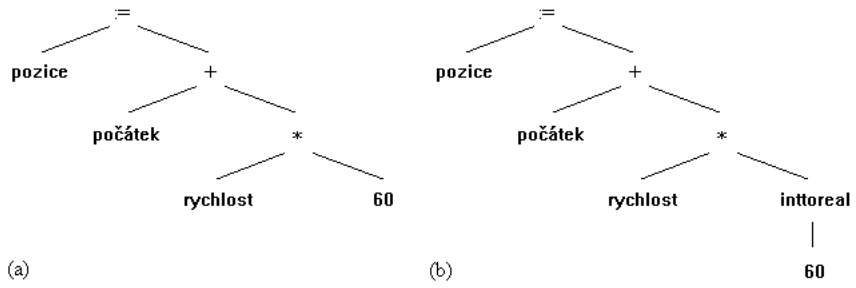
Podobným způsobem jsou defnovány příkazy jazyka, jako např.:

$$\text{příkaz} \; -> \; \text{identifikátor} := \text{výraz} \tag{4.6}$$
$$\text{příkaz} \; -> \; \text{while}\ (\ \text{výraz}\ )\ \text{do příkaz (1.2)} \tag{4.7}$$
$$\text{příkaz} \; -> \; \text{if}\ (\ \text{výraz}\ )\ \text{then příkaz} \tag{4.8}$$

Dělení na lexikální a syntaktickou analýzu je dosti volné. Obvykle vybíráme takové rozdělení, které zjednodušuje činnost analýzy. Jedním z faktorů, které přitom uvažujeme, je to, zda jsou konstrukce zdrojového jazyka regulární nebo ne. Lexikální jednotky lze obvykle popsat jako regulární množiny, zatímco konstrukce vytvořené z lexikálních jednotek již vyžadují obecnější přístupy.

Typickou regulární konstrukcí jsou identifkátory, popsané obvykle jako posloupnosti písmen a číslic začínající písmenem. Běžně rozpoznáváme identifikátory jednoduchým prohlížením vstupního textu, v němž očekáváme znak, který není písmeno ani číslice, a potom seskupíme všechna písmena a číslice nalezené až do tohoto místa do lexikální jednotky pro identifikátor. Znaky takto shromážděné zaznamenáme do tabulky (tabulky symbolů) a odstraníme je ze vstupu tak, aby mohlo pokračovat zpracování dalšího symbolu.

Tento způsob lineárního prohledávání na druhé straně není dostatečný pro analýzu výrazů nebo příkazů. Nemůžeme například jednoduše kontrolovat dvojice závorek nebo klíčových slov `begin` a `end` v příkazech bez zavedení jakéhosi druhu hierarchické struktury na vstupu.



Obrázek 4.2: Syntaktický strom

Derivační strom na obr. 4.1 popisuje syntaktickou strukturu vstupu, ale obsahuje informace, které nejsou důležité pro další průběh překladu. Mnohem běžnější vnitřní reprezentaci

této syntaktické struktury dává syntaktický strom na obrázku 4.2(a). Syntaktický strom je zhuštěnou reprezentací derivačního stromu; operátory v něm vystupují jako vnitřní uzly a operandy těchto operátorů jsou následníky jejich příslušných uzlů.

Po syntaktické analýze většinou následuje v interpreterech i v překladačích analýza sématická. V tomto projektu je ovšem nevyužita, nebo alespoň v té podobě jak je chápána. Byla by ale škoda ji zde neuvést. Pokračuje v popisování příkladu uvedeného na předchozích stránkách, který ne zcela koresponduje s problematikou LaTeXu, ale k vysvětlení problematiky se hodí lépe.

Fáze sémantické analýzy zpracovává především informace, které jsou uvedeny v deklaracích, ukládá je do vnitřních datových struktur a na jejich základě provádí sémantickou kontrolu příkazů a výrazů v programu. K identifikaci operátorů a operandů těchto výrazů a příkazů využívá hierarchickou strukturu, určenou ve fázi syntaktické analýzy.

Důležitou složkou sémantické analýzy je typová kontrola. Kompilátor zde kontroluje, zda všechny operátory mají operandy povolené specifikací zdrojového jazyka. Mnoho definic programovacích jazyků například vyžaduje, aby kompilátor hlásil chybu, kdykoliv je reálné číslo použito jako index pole. Specifikace jazyka však může dovolit některé implicitní transformace operandů, například při aplikaci binárního aritmetického operátoru na celočíselný a reálný operand. V tomto případě může kompilátor požadovat konverzi celého čísla na reálné.

Jsou-li např. všechny proměnné v našem ukázkovém příkazu reálné, je třeba provést konverzi celočíselné konstanty 60 na reálnou, jak znázorňuje obr. 4.2(b). V tomto případě je rovněž možné typovou konverzi provést přímo a konstantu 60 nahradit hodnotou 60.0.

Po tomto úvodu následuje analýza dvou základních přístupů k syntaktické analýze, přístup zdola-nahoru (bottom-up) a přístup zhora-dolů (top-down). Tyto přístupy jsou zhodnoceny z hlediska možnosti jejich využití v tomto projektu.

4.1 Top-down analýza

Syntaktický analyzátor shora dolů vytváří derivační strom věty bezkontextového jazyka od počátečního symbolu (kořene stromu) a postupně doplňuje hrany a uzly směrem dolů a zleva doprava. Syntaktická analýza je konstrukce derivačního stromu pomocí zásobníkového automatu, který využívá rozkladovou tabulku.

Je taková gramatika $G = (N, T, P, S)$, kde:

- N je konečná množina neterminálních symbolů
- T je konečná množina terminálních symbolů ($N \cap T = \emptyset$)
- P je konečná množina pravidel zapisovaných ve tvaru $A \rightarrow \alpha$, kde $A \in N$, $\alpha \in (N \cup T)*$
- S je počáteční (startovací) symbol gramatiky

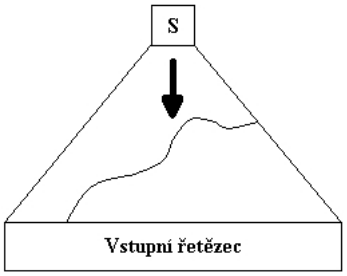
Levou stranou pravidla $A \rightarrow \alpha$ je chápán neterminál A, pravou stranou řetězec α.

Několik pravidel se stejnou levou stranou se zapisuje též zkráceně ve tvaru

$$A \rightarrow \alpha_1 | \alpha_2 | \ldots | \alpha_k.$$

Pro vytvoření LL syntaktického analyzátoru je nejprve nutno spočítat množiny $EMPTY$, $FIRST$, $FOLLOW$ a $PREDICT$. Následuje stručný postup vytvoření těchto množin.

- $FIRST(\alpha)$ je množina terminálů, které se můžou vyskytnou na začátku řetězců derivovaných z α
- $FOLLOW(A)$ je množina terminálů, které se mohou vyskytnou za neterminálem A Vstupem je bezkontextová gramatika $G = (N, T, P, S)$ a výstupem jsou množiny $FOLLOW(A)$ pro každé A z N
  1. položíme $FOLLOW(S) = \{\varepsilon\}$, $FOLLOW(X) = \emptyset$ pro $\forall X \in N, X \neq S$
  2. pro všechny výskyty nonterminálních symbolů na pravých stranách pravidel provedeme:
     (a) má-li pravidlo tvar $X \rightarrow \alpha Y \beta$, pak $FOLLOW(Y) = FOLLOW(Y) \bigcup (FIRST(\beta) - \{\varepsilon\})$
     (b) má-li pravidlo tvar $X \rightarrow \alpha Y \beta$ a platí-li $\varepsilon \in FIRST(\varepsilon)$, pak $FOLLOW(Y) = FOLLOW(Y) \bigcup FOLLOW(X)$
  3. opakujeme krok 2, pokud se změnila alespoň jedna množina $FOLLOW(X)$
- $PREDICT(A \rightarrow \alpha)$
  - Pokud $EMPTY(\alpha) = \{\varepsilon\}$ pak $PREDICT(A \rightarrow \alpha) = FIRST(\alpha) \bigcup FOLLOW(A)$
  - Pokud $EMPTY(\alpha) = \emptyset$ pak $PREDICT(A \rightarrow \alpha) = FIRST(\alpha)$
- syntaktický analyzátor pro LL(1) gramatiku je upravený zásobníkový automat, který pro výběr pravé strany pravidla při operaci expanze používá tzv. rozkladovou tabulku. Formálně to je zobrazení definované na kartézském součinu $N \times T \bigcup \varepsilon$
- hodnotou $M(A, a)$je číslo pravidla, které se má použít pro expanzi neterminálu A a dopředu přečteném vstupním symbolu a značka chyby.



Obrázek 4.3: Syntaktická analýza shora dolů

Vytvoření rozkladové tabulky pro LL(1) gramatiku, kde vstupem je LL(1) gramatika $G = (N, T, P, S)$ výstupem je rozkladová tabulka M pro gramatiku G :

1. je-li $A \rightarrow \alpha$ i-té pravidlo $\in P$, pak $M(A, a) = i$ pro $\forall a \in FIRST(\alpha) - \varepsilon$
2. je-li $A \rightarrow \alpha$ i-té pravidlo $\in P$ a $\alpha \in FIRST(\alpha)$, pak $M(A, b) = i$ pro $\forall b \in FOLLOW(A)$

3. $M(A, a) = error$ ve všech ostatních případech

Left to right, produkuje Leftmost derivation. LL(k), k je počet look-ahead tokenů. Bezkontextová gramatika je LL(1) pokud pro každou dvojici pravidel $A \to \alpha|\beta$ platí:

1. $FIRST(\alpha) \bigcap FIRST(\beta) = \emptyset$
2. jestliže $\varepsilon \in FIRST(\alpha) \wedge \varepsilon \notin FIRST(\beta)$, pak $FOLLOW(A) \bigcap FIRST(\beta) = \emptyset$

Následují další úpravy pravidel, které sice nejsou použity v této diplomové práci, ale byla by škoda je zde nezmínit.

Odstranění levé rekurze -

Rekurzivita v BG může vzniknout tak, že se na pravé straně pravidla objeví neterminální symbol ze strany levé, tedy $A \to aA$ (pravá) nebo $A \to Aa$ (levá). Pravidla $A \to A\alpha_1|A\alpha_2|\beta_1|\beta_2$ se převedou na pravidla následující (zaveden nový nonterminál A'):

- $A \to \beta_1 A'|\beta_2 A'|\beta_1|\beta_2$
- $A' \to \alpha_1 A'|\alpha_2 A'|\alpha_1|\alpha_2$

Levá faktorizace - levá faktorizace odstraňuje kolize FIRST-FIRST u pravidel s pravými stranami začínajícími týmž řetězcem pravidla typu $A \to \alpha\alpha_1|\alpha\alpha_2|\ldots|\alpha\alpha_n$ je nahrazen podle vzorců. Je přitom zaveden nový nonterminál:

- $A \to \alpha A'$
- $A' \to \alpha_1|\alpha_2|\ldots|\alpha_n$

Rohová substituce - řešení pro pravidla s kolizí FIRST-FIRST, která nelze přímo faktorizovat:

nechť pravidla typu $B \to \beta_1|\beta_2|\ldots|\beta_m$ jsou všechna B-pravidla gramatiky. Pak pravidlo $A \to B\alpha$ lze nahradit podle vzorce $A \to \beta_1\alpha|\beta_2\alpha|\ldots|\beta_m\alpha$

Pohlcení terminálu - používá se na odstranění konfliktu First-Follow.

Existuje pravidlo $A \to \alpha Ba\beta$ a pravidla $B \to \alpha_1|\alpha_2|\ldots|\alpha_m$. Pak první pravidlo nahradíme pravidly $A \to \alpha[Ba]\beta$ a $[Ba] \to \alpha_1 a|\alpha_2 a|\ldots|\alpha_m a$.

Extrakce pravého kontextu - opakované dosazování za nonterminály.

Pohlcení řetězce Existuje pravidlo $A \to \alpha B\beta\gamma$ a pravidla $B \to \alpha_1|\alpha_2|\ldots|\alpha_m$. Pak první pravidlo je nahrazeno pravidly $A \to \alpha[B\beta]\gamma$ a $[B\beta] \to \alpha_1\beta|\alpha_2\beta|\ldots|\alpha_m\beta$.

Realizace LL(1) syntaktického analyzátoru pomocí rekurzivního sestupu spočívá ve vytvoření samostatných procedur pro analýzu každého nonterminálního symbolu. Přitom tělo každé procedury je tvořeno řídící strukturou, která větví analýzu na základě rozkladové LL tabulky. Každá alternativa řídící struktury je tvořena posloupností příkazů volání procedur, které odpovídají symbolům příslušné pravé strany pravidla. Analýza se spustí voláním procedury odpovídající počátečnímu symbolu gramatiky (musí být již přečten lexikálním analyzátorem první lexikální element). Tato metoda se zdá být nejvhodnější pro využití v případě interpreteru LaTeXu, její rekurzivní volání procedur může značně usnadnit analýzu LaTeXových prostředí, které mohou být v sobě rekurzivně zanořené.

4.2 Bottom-up analýza

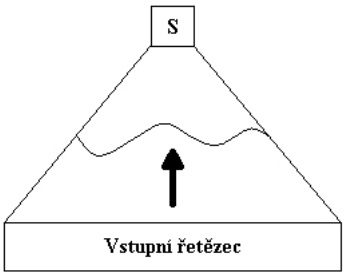
LR syntaktická analýza probíhá odspoda nahoru, čte symboly odleva a vytváří pravou derivaci. Zásobníkový automat pro syntaktickou analýzu zdola nahoru:

- $\delta(q, a, \varepsilon) = \{(\text{q, a})\}$ pro všechna $a \in T$ (přesun - přesun vstupního symbolu na vrchol zásobníku)
- $\delta(q, \varepsilon, \alpha) = \{(q, A) : A \to \alpha \in P\}$ (redukce - náhrada pravé strany levou stranou)
- $\delta(q, \varepsilon, \#S) = \{(r, \varepsilon)\}$ (přijetí)
- vrchol zásobníku je vpravo, jeho iniciálním stavem je #.

Pro konstrukci LR(0) syntaktického analyzátoru je nutno uvažovat následující algoritmus:

1. Vytvoříme kolekci množin LR(0) položek. LR(0) položka je ve tvaru $A \to \alpha \cdot \beta$, kde tečka udává pozici. LR(0) položka reprezentuje zásobníkový symbol, tedy kolekce LR(0) položek je množina všech stavů, ve kterých se může analyzátor nacházet.
   (a) Do počáteční množiny # je vložena položka $S' \to \cdot S$.
   (b) Je vytvořen uzávěr počáteční množiny: Když množina obsahuje. $A \to \alpha \cdot B\gamma$, tak do této množiny je přidáno $B \to \cdot\delta$. Je tedy provedena expanze neterminálu, který následuje za tečkou
   (c) Pro každé pravidlo $A \to \alpha \cdot X\beta$ je vytvořen následník pro $X : MX = \{A \to \alpha X \cdot \beta\}$.
   (d) Je vytvořen uzávěr nad všemi množinami.
   (e) Takto se pokračuje, dokud nevznikají další množiny a následníci. Každá množina z kolekce se tedy vyznačuje tím, že všechny LR(0) položky v ní mají před tečkou tentýž symbol nebo je tečka na začátku pravidla.
2. Je nutno vytvořit tabulku akcí a přechodů, která bude řídit syntaktický analyzátor.
   (a) Tato tabulka má pro každou množinu z kolekce množin LR(0) položek jeden řádek.
   (b) Tabulka akcí má jeden sloupec a tabulka přechodů má sloupec pro každý terminál a neterminál.
   (c) Tabulka akcí je vytvořena tak, že pokud v odpovídající množině LR(0) položek je pravidlo $A \to \alpha \cdot a\beta$, je zapsán do tabulky akcí přesun.
   (d) Pokud obsahuje pravidlo $A \to \alpha\cdot$, je zapsána redukce podle tohoto pravidla.
   (e) Při konstrukci tabulky akcí jsou tedy použity pouze LR(0) položky, které mají tečku před terminálem nebo na konci pravidla.
   (f) Tabulka přechodů je zkonstruována z toho, jak byla vytvářena kolekce množin LR(0) položek.

Pokud je v řádku tabulky akcí redukce, potom řádek tabulky přechodů neobsahuje žádnou položku - záznam tam je jen tehdy, pokud akcí je přesun. Je možno tedy použít jedinou tabulku s jedním sloupcem, která bude obsahovat položky přesun a redukce. Tím je sice



Obrázek 4.4: Syntaktická analýza zdola nahoru

snížen počet získatelných informací, ale syntaktická analýza proběhne zcela identicky. Tato tabulka s jedním sloupcem je ale nutná pro činnost LR(0) syntaktického analyzátoru, jenž nečte dopředu žádný symbol ze vstupu.

LR(0) gramatika definována tak, že pokud množina z kolekce množin LR(0) položek obsahuje pravidlo $A \rightarrow \alpha\cdot$., pak nesmí obsahovat pravidlo $B \rightarrow \ \beta\cdot$ ani $B \rightarrow \beta \cdot \gamma$, kde γ začíná terminálem. Množiny tedy nesmí obsahovat kolizi redukce-redukce ani kolizi přesun-redukce.

Při provádění syntaktické analýzy je využita informace z konstruovaných tabulek. Pokud je prováděna redukce, je nutno určit, jaký konkrétní neterminál bude umístěn na zásobník (S1, S2, ...). To je provedeno tak, že ze zásobníku je odstraněna celá pravá strana nahrazovaného pravidla a podle tabulky přechodů, konkrétně řádku pro symbol na zásobníku a sloupce pro neterminál, který bude na zásobník vložen.

4.3 Shrnutí

Výše byly uvedeny různé metody syntaktické analýzy. Pro syntaktickou analýzu LaTeXu je nejvhodnější využití jedné, maximálně dvou metod. Díky struktuře tohoto jazyka se jeví jako nejvhodnější metoda rekurzivním sestupem, která umožňuje poměrně snadnou implementaci a její analýza rekurzivně zanořených prostředí LaTeXu se jeví jako nejúčinnější. V následující podkapitole jsou uvedeny pravidla, které byly využity při syntaktické analýze implementované v interpreteru. Konstrukce výše zmiňovaných množin $EMPTY$, $FIRST$, $FOLLOW$ a $PREDICT$ není v diplomové práci uváděna, neboť celkový počet pravidel (použitých i nevyužitých) dosahuje téměr 950, což znemožňuje efektivní ruční tvorbu těchto množin a přechodových tabulek. Využití nástrojů jako je YACC či YARD bylo zavrženo.

LR analýza by mohla být v implementaci interpreteru využita spíše jen okrajově, nabízí se možnost analýzy argumentů určitých prostředí, například `tabular` nebo příkazu pro vložení obrázku `\includegraphics`. V této verzi ale není zamýšleno tyto prostředí a příkazy přímo interpretovat. Nyní budou převáděny do formátu gif pomocí skriptu `textogif`. Proto je LR syntaktická analýza momentálně zbytečná a proto nevyužita. V příštích verzích by již ale mohla být implementována.

4.4 Syntaktická pravidla

V hranatých závorkách (<, >) jsou uvedeny nonterminály, slova bez závorek jsou chápány jako terminály.

<S> $\Longrightarrow$ <class> <package> <document>
<class> $\Longrightarrow$ `\documentclass` <volby> <styl>

<volby> $\Longrightarrow$ [<in-volby>]
<volby> $\Longrightarrow$ ε
<in-volby> $\Longrightarrow$ <mira> $\|$ a4paper $\|$ a5paper $\|$ b5paper $\|$ letterpaper $\|$ executivepaper $\|$ landscape $\|$ twoside $\|$ oneside $\|$ twocolumn <aux-volby>
<aux-volby> $\Longrightarrow$,<in-volby>
<aux-volby> $\Longrightarrow$ ε
<styl> $\Longrightarrow$ book $\|$ article $\|$ letter $\|$ report
<package> $\Longrightarrow$ `\usepackage` <volby-package> name-package <package>
<package> $\Longrightarrow$ ε
<volby-package> $\Longrightarrow$ [<pack-volby>]
<pack-volby> $\Longrightarrow$ <odsazeni><next>
<odsazeni> $\Longrightarrow$ left=<mira> $\|$ right= <mira> $\|$ top= <mira> $\|$ bottom= <mira> $\|$ text={<mira> , <mira>}
<odsazeni> $\Longrightarrow$
<next> $\Longrightarrow$,<odsazeni>
<next> $\Longrightarrow$ ε

<document> $\Longrightarrow$ `\begin`{document} <body> `\end`{document}

<body> $\Longrightarrow$ $ <mat-text> $ <body>

<body> $\Longrightarrow$ {}<body>
<body> $\Longrightarrow$ a-z0-9<body>
<body> $\Longrightarrow$ `\,` <body>
<body> $\Longrightarrow$ `-` <body>
<body> $\Longrightarrow$ `--` <body>
<body> $\Longrightarrow$ `---` <body>
<body> $\Longrightarrow$ `\%` <body>
<body> $\Longrightarrow$ `\promile` <body>
<body> $\Longrightarrow$ `\{` <body>
<body> $\Longrightarrow$ `\}` <body>
<body> $\Longrightarrow$ `\uv`<popis> <body>
<body> $\Longrightarrow$ `\clqq` <body>
<body> $\Longrightarrow$ `\crqq` <body>
<body> $\Longrightarrow$ `\textquotedblletf` <body>
<body> $\Longrightarrow$ `\textquotedblright` <body>
<body> $\Longrightarrow$ `\frqq` <body>
<body> $\Longrightarrow$ `\flqq` <body>
<body> $\Longrightarrow$ `\textquoteleft` <body>
<body> $\Longrightarrow$ `\textquoteright` <body>
<body> $\Longrightarrow$ `\quotesinglbase` <body>
<body> $\Longrightarrow$ `\S` <body>
<body> $\Longrightarrow$ `\&` <body>
<body> $\Longrightarrow$ `\dots` <body>
<body> $\Longrightarrow$ `\$` <body>

<body> $\Longrightarrow$ `\#` <body>
<body> $\Longrightarrow$ `\dag` <body>
<body> $\Longrightarrow$ `\ddag` <body>
<body> $\Longrightarrow$ `\P` <body>
<body> $\Longrightarrow$ `\copyright` <body>
<body> $\Longrightarrow$ `\pounds` <body>
<body> $\Longrightarrow$ `\TeX` <body>
<body> $\Longrightarrow$ `\LaTeX` <body>
<body> $\Longrightarrow$ `\LaTeXe` <body>
<body> $\Longrightarrow$ `\today` <body>
<body> $\Longrightarrow$ `\centering` <body>
<body> $\Longrightarrow$ `\raggedleft` <body>
<body> $\Longrightarrow$ `\raggedright` <body>
<body> $\Longrightarrow$ `\'`{znak}<body>
<body> $\Longrightarrow$ `\‘`{znak}<body>
<body> $\Longrightarrow$ `\^`{znak}<body>
<body> $\Longrightarrow$ `\"`{znak}<body>
<body> $\Longrightarrow$ `\~`{znak}<body>
<body> $\Longrightarrow$ `\=`{znak}<body>
<body> $\Longrightarrow$ `\.`{znak}<body>
<body> $\Longrightarrow$ `\u`{znak}<body>
<body> $\Longrightarrow$ `\v`{znak}<body>
<body> $\Longrightarrow$ `\H`{znak}<body>
<body> $\Longrightarrow$ `\t`{znak}<body>
<body> $\Longrightarrow$ `\r`{znak}<body>
<body> $\Longrightarrow$ `\c`{znak}<body>
<body> $\Longrightarrow$ `\d`{znak}<body>
<body> $\Longrightarrow$ `\b`{znak}<body>
<body> $\Longrightarrow$ `\k`{znak}<body>
<body> $\Longrightarrow$ `\i` <body>
<body> $\Longrightarrow$ `\j` <body>
<body> $\Longrightarrow$ `\oe` <body>
<body> $\Longrightarrow$ `\OE` <body>
<body> $\Longrightarrow$ `\ae` <body>
<body> $\Longrightarrow$ `\AE` <body>
<body> $\Longrightarrow$ `\aa` <body>
<body> $\Longrightarrow$ `\AA` <body>
<body> $\Longrightarrow$ `\o` <body>
<body> $\Longrightarrow$ `\O` <body>
<body> $\Longrightarrow$ `\l` <body>
<body> $\Longrightarrow$ `\L` <body>
<body> $\Longrightarrow$ `\ss` <body>
<body> $\Longrightarrow$ `?’` <body>
<body> $\Longrightarrow$ `!’` <body>
<body> $\Longrightarrow$ {<body> } <body>
<body> $\Longrightarrow$ {<in-text> <body> }
<body> $\Longrightarrow$ <in-text> <body>
<in-text> $\Longrightarrow$ `\tiny`

<in-text> $\Longrightarrow$ `\scriptsize`
<in-text> $\Longrightarrow$ `\footnotesize`
<in-text> $\Longrightarrow$ `\small`
<in-text> $\Longrightarrow$ `\normalsize`
<in-text> $\Longrightarrow$ `\large`
<in-text> $\Longrightarrow$ `\Large`
<in-text> $\Longrightarrow$ `\LARGE`
<in-text> $\Longrightarrow$ `\huge`
<in-text> $\Longrightarrow$ `\Huge`
<in-text> $\Longrightarrow$ `\rmfamily`
<in-text> $\Longrightarrow$ `\sffamily`
<in-text> $\Longrightarrow$ `\ttfamily`
<in-text> $\Longrightarrow$ `\mdseries`
<in-text> $\Longrightarrow$ `\bfseries`
<in-text> $\Longrightarrow$ `\upshape`
<in-text> $\Longrightarrow$ `\itshape`
<in-text> $\Longrightarrow$ `\slshape`
<in-text> $\Longrightarrow$ `\scshape`
<in-text> $\Longrightarrow$ `\rm`
<in-text> $\Longrightarrow$ `\it`
<in-text> $\Longrightarrow$ `\bf`
<in-text> $\Longrightarrow$ `\sl`
<in-text> $\Longrightarrow$ `\sf`
<in-text> $\Longrightarrow$ `\sc`
<in-text> $\Longrightarrow$ `\tt`
<in-text> $\Longrightarrow$ `\em`

<body> $\Longrightarrow$ `\textrm` <popis> <body>
<body> $\Longrightarrow$ `\textsf` <popis> <body>
<body> $\Longrightarrow$ `\texttt` <popis> <body>
<body> $\Longrightarrow$ `\textmd` <popis> <body>
<body> $\Longrightarrow$ `\textbf` <popis> <body>
<body> $\Longrightarrow$ `\textup` <popis> <body>
<body> $\Longrightarrow$ `\textit` <popis> <body>
<body> $\Longrightarrow$ `\textsl` <popis> <body>
<body> $\Longrightarrow$ `\textsc` <popis> <body>
<body> $\Longrightarrow$ `\emph` <popis> <body>
<body> $\Longrightarrow$ `\/` <body>
<body> $\Longrightarrow$ `\,` <body>
<body> $\Longrightarrow$ `\@` <body>
<body> $\Longrightarrow$ `~` <body>
<body> $\Longrightarrow$ `\frenchspacing` <body>
<body> $\Longrightarrow$ `\nofrenchspacing` <body>
<body> $\Longrightarrow$ `\stretch` <prumer> <body>
<body> $\Longrightarrow$ <nastav-font> `\selectfont` <body>
<nastav-font> $\Longrightarrow$ `\fontfamily` <rodina> <nastav-font>
<nastav-font> $\Longrightarrow$ `\fontencoding` <kodovani> <nastav-font>
<nastav-font> $\Longrightarrow$ `\fontseries` <vaha> <nastav-font>

<nastav-font> $\Longrightarrow$ `\fontshape` <tvar> <nastav-font>
<nastav-font> $\Longrightarrow$ `\fontsize` <prumer><prumer> <nastav-font>
<nastav-font> $\Longrightarrow \varepsilon$
<rodina> $\Longrightarrow$ {cmr}
<rodina> $\Longrightarrow$ {cmss}
<rodina> $\Longrightarrow$ {cmtt}
<rodina> $\Longrightarrow$ {cmm}
<rodina> $\Longrightarrow$ {cmsy}
<rodina> $\Longrightarrow$ {cmex}
<rodina> $\Longrightarrow$ {pag}
<rodina> $\Longrightarrow$ {pbk}
<rodina> $\Longrightarrow$ {pcr}
<rodina> $\Longrightarrow$ {phv}
<rodina> $\Longrightarrow$ {ppl}
<rodina> $\Longrightarrow$ {pnc}
<rodina> $\Longrightarrow$ {ptm}
<rodina> $\Longrightarrow$ {pzc}

<kodovani> $\Longrightarrow$ {0T1}
<kodovani> $\Longrightarrow$ {T1}
<kodovani> $\Longrightarrow$ {0ML}
<kodovani> $\Longrightarrow$ {0MS}
<kodovani> $\Longrightarrow$ {0MX}
<kodovani> $\Longrightarrow$ {U}
<kodovani> $\Longrightarrow$ {Lcislo}

<vaha> $\Longrightarrow$ {m}
<vaha> $\Longrightarrow$ {b}
<vaha> $\Longrightarrow$ {bx}
<vaha> $\Longrightarrow$ {sb}
<vaha> $\Longrightarrow$ {c}

<tvar> $\Longrightarrow$ {n}
<tvar> $\Longrightarrow$ {it}
<tvar> $\Longrightarrow$ {sl}
<tvar> $\Longrightarrow$ {sc}

<body> $\Longrightarrow$ `\normalfont` <body>
<body> $\Longrightarrow$ `\usefont`<kodovani><rodina><vaha><tvar> <body>
<body> $\Longrightarrow$ `\hspace`<hvezda><prumer> <body>
<body> $\Longrightarrow$ `\hspace`<hvezda><del-reg> <body>
<hvezda> $\Longrightarrow$ *
<hvezda> $\Longrightarrow \varepsilon$
<del-reg> $\Longrightarrow$ {<nasobek><registry>}
<del-reg> $\Longrightarrow$ {`\fill`}
<nasobek> $\Longrightarrow$ cislo
<nasobek> $\Longrightarrow$ -
<nasobek> $\Longrightarrow \varepsilon$

<body> $\Longrightarrow$ `\vspace` <hvezda><prumer> <body>
<body> $\Longrightarrow$ `\hspace` <hvezda><del-reg> <body>
<body> $\Longrightarrow$ `\smallskip` <body>
<body> $\Longrightarrow$ `\medskip` <body>
<body> $\Longrightarrow$ `\bigskip` <body>
<body> $\Longrightarrow$ `\hfill` <body>
<body> $\Longrightarrow$ `\vfill` <body>
<body> $\Longrightarrow$ `\dotfill` <body>
<body> $\Longrightarrow$ `\hrulefill` <body>
<body> $\Longrightarrow$ `\underline` <popis> <body>
<body> $\Longrightarrow$ `%` <body>
<body> $\Longrightarrow$ `\\` <hvezda><prumer-nepovinny>
<prumer-nepovinny> $\Longrightarrow$ [mira]
<prumer-nepovinny> $\Longrightarrow$ ε
<body> $\Longrightarrow$ `\-` <body>
<body> $\Longrightarrow$ `\par` <body>

<body> $\Longrightarrow$ `\newlength` <reg> <body>
<body> $\Longrightarrow$ `\setlength` <reg><reg2> <body>
<body> $\Longrightarrow$ `\addtolength` <reg><reg2> <body>
<body> $\Longrightarrow$ `\settowidth` <reg><popis> <body>
<body> $\Longrightarrow$ `\settoheigth` <reg><popis> <body>
<body> $\Longrightarrow$ `\settodepth` <reg><popis> <body>
<reg> $\Longrightarrow$ { <registry>}
<reg2> $\Longrightarrow$ { <mira>}
<reg2> $\Longrightarrow$ { <registry>}
<registry> $\Longrightarrow$ `\voffset`
<registry> $\Longrightarrow$ `\hoffset`
<registry> $\Longrightarrow$ `\oddsidemargin`
<registry> $\Longrightarrow$ `\evensidemargin`
<registry> $\Longrightarrow$ `\topmargin`
<registry> $\Longrightarrow$ `\headheight`
<registry> $\Longrightarrow$ `\headsep`
<registry> $\Longrightarrow$ `\textwidth`
<registry> $\Longrightarrow$ `\textheigth`
<registry> $\Longrightarrow$ `\marginparwidth`
<registry> $\Longrightarrow$ `\marginparsep`
<registry> $\Longrightarrow$ `\marginparpush`
<registry> $\Longrightarrow$ `\pageheight`
<registry> $\Longrightarrow$ `\footskip`
<registry> $\Longrightarrow$ `\pagewidth`
<registry> $\Longrightarrow$ `\leftskip`
<registry> $\Longrightarrow$ `\rightskip`
<registry> $\Longrightarrow$ `\parskip`
<registry> $\Longrightarrow$ `\parindent`
<registry> $\Longrightarrow$ `\linewidth`
<registry> $\Longrightarrow$ `\baselineskip`
<registry> $\Longrightarrow$ <text>

<body> ⟹ \newpage <body>
<body> ⟹ \clearpage <body>
<body> ⟹ \cleardoublepage <body>
<body> ⟹ \pagebreak <N> <body>
<body> ⟹ \nopagebreak <N> <body>
<body> ⟹ \samepage <body>
<body> ⟹ \enlargethispage <hvezda><prumer>
<body> ⟹ \verb <hvezda><znak> <body> <znak> <body>
<body> ⟹ \footnote <N><popis> <body>
<body> ⟹ \footnotemark <nepovinne> <body>
<body> ⟹ \footnotetext <nepovinne><povinne> <body>
<body> ⟹ \footnoterule <body>
<body> ⟹ \frontmatter <body>
<body> ⟹ \mainmatter <body>
<body> ⟹ \backmatter <body>
<body> ⟹ \part <hvezda><nepov-text><pov-text> <body>
<body> ⟹ \chapter <hvezda><nepov-text><pov-text> <body>
<body> ⟹ \section <hvezda><nepov-text><pov-text> <body>
<body> ⟹ \subsection <hvezda><nepov-text><pov-text> <body>
<body> ⟹ \subsubsection <hvezda><nepov-text><pov-text> <body>
<body> ⟹ \paragraph <hvezda><nepov-text><pov-text> <body>
<body> ⟹ \subparagraph <hvezda><nepov-text><pov-text> <body>
<nepov-text> ⟹ [text]
<pov-text> ⟹ {text}
<body> ⟹ \twocolumn <body>
<body> ⟹ \onecolumn <body>
<body> ⟹ \flushbottom <body>
<body> ⟹ \raggedbottom <body>
<body> ⟹ \pagestyle <pg-styl> <body>
<pg-styl> ⟹ {plain}
<pg-styl> ⟹ {empty}
<pg-styl> ⟹ {headings}
<pg-styl> ⟹ {myheadings}
<body> ⟹ \pagenumbering <pg-cislo> <body>
<pg-cislo> ⟹ {\roman}
<pg-cislo> ⟹ {\Roman}
<pg-cislo> ⟹ {\arabic}
<pg-cislo> ⟹ {\Arabic}
<pg-cislo> ⟹ {\alph}
<pg-cislo> ⟹ {\Alph}

<body> ⟹ \color <pov-barva> <body>
<body> ⟹ \colorbor <pov-barva><popis> <body>
<body> ⟹ \definecolor <popis><barva><procento> <body>
<body> ⟹ \fcolorbox <pov-barva><pov-barva><popis> <body>
<body> ⟹ \pagecolor <pov-barva> <body>
<body> ⟹ \textcolor <pov-barva><popis> <body>

<pov-barva> $\Longrightarrow$ {black}
<pov-barva> $\Longrightarrow$ {white}
<pov-barva> $\Longrightarrow$ {red}
<pov-barva> $\Longrightarrow$ {green}
<pov-barva> $\Longrightarrow$ {blue}
<pov-barva> $\Longrightarrow$ {yellow}
<pov-barva> $\Longrightarrow$ {cyan}
<pov-barva> $\Longrightarrow$ {magenta}
<barva2> $\Longrightarrow$ [<bar>]
<barva> $\Longrightarrow$ {<bar>}
<bar> $\Longrightarrow$ rgb
<bar> $\Longrightarrow$ gray
<bar> $\Longrightarrow$ cmyk

<body> $\Longrightarrow$ `\marginpar`<nepovinne><popis> <body>
<body> $\Longrightarrow$ `\noindent` <body>
<body> $\Longrightarrow$ `\newpage` <body>
<body> $\Longrightarrow$ `\newcommand` {text}{text} <body>
<body> $\Longrightarrow$ `\renewcommand` {text}{text} <body>
<body> $\Longrightarrow$ `\providecommand` {text}[text][text]{text} <body>
<body> $\Longrightarrow$ `\ensuremath` {<mat-text>} <body>
<body> $\Longrightarrow$ `\def` <text> <param> <popis> <body>
<param> $\Longrightarrow$ #cislo <param2>
<param2> $\Longrightarrow$ #cislo <param2>
<param2> $\Longrightarrow$ ε
<body> $\Longrightarrow$ `\newcounter` {<text>}
<body> $\Longrightarrow$ `\newenviroment` {text}[][]{text}{text} <body>
<body> $\Longrightarrow$ `\DeclareFontFamily` {}{}{} <body>
<body> $\Longrightarrow$ `\DeclareFontShape` {}{}{}{}{}{} <body>
<body> $\Longrightarrow$ `\DeclareSymbolFont` {}{}{}{}{} <body>
<body> $\Longrightarrow$ `\DeclareMathSymbol` {}{}{}{} <body>
<body> $\Longrightarrow$ `\relax` <body>
<body> $\Longrightarrow$ `\setcounter` <counter2><pocet> <body>
<body> $\Longrightarrow$ `\addtocounter` <counter2><pocet> <body>
<body> $\Longrightarrow$ `\newcounter` <counter2><counter3> <body>
<counter3> $\Longrightarrow$ [<counter>]
<counter2> $\Longrightarrow$ {<counter>}
<counter> $\Longrightarrow$ `part`
<counter> $\Longrightarrow$ `chapter`
<counter> $\Longrightarrow$ `section`
<counter> $\Longrightarrow$ `subsection`
<counter> $\Longrightarrow$ `subsubsection`
<counter> $\Longrightarrow$ `paragraph`
<counter> $\Longrightarrow$ `subparagraph`
<counter> $\Longrightarrow$ `page`
<counter> $\Longrightarrow$ `equation`
<counter> $\Longrightarrow$ `figure`
<counter> $\Longrightarrow$ `table`

<counter> $\Longrightarrow$ `footnote`
<counter> $\Longrightarrow$ `mpfootnote`
<counter> $\Longrightarrow$ `enumi`
<counter> $\Longrightarrow$ `enumii`
<counter> $\Longrightarrow$ `enumiii`
<counter> $\Longrightarrow$ `enumiv`
<counter> $\Longrightarrow$ <text>
<body> $\Longrightarrow$ `\value` <counter2> <body>
<body> $\Longrightarrow$ `\stepcounter` <counter2> <body>
<body> $\Longrightarrow$ `\refstepcounter` <counter2> <body>
<body> $\Longrightarrow$ `\the` <text> <body>
<body> $\Longrightarrow$ `\label` <popis> <body>
<body> $\Longrightarrow$ `\ref` <popis> <body>
<body> $\Longrightarrow$ `\pageref` <popis> <body>
<body> $\Longrightarrow$ `\makebox` <n-prumer><poz-box><popis> <body>
<body> $\Longrightarrow$ `\makebox` <box-reg><poz-box><popis> <body>
<box-reg> $\Longrightarrow$ [cislo<b-reg>]
<box-reg> $\Longrightarrow$ ε
<b-reg> $\Longrightarrow$ `\height`
<b-reg> $\Longrightarrow$ `\depth`
<b-reg> $\Longrightarrow$ `\totalheight`
<b-reg> $\Longrightarrow$ `\width`
<poz-box> $\Longrightarrow$ [l ‖ r ‖ s]
<poz-box> $\Longrightarrow$ ε
<body> $\Longrightarrow$ `\framebox` <n-prumer><poz-box><popis> <body>
<body> $\Longrightarrow$ `\framebox` <box-reg><poz-box><popis> <body>
<body> $\Longrightarrow$ `\parbox` <parbox-poz><poz-box><poz-parbox><prumer><popis><body>
<parbox-poz> $\Longrightarrow$ [t ‖ b]
<parbox-poz> $\Longrightarrow$ ε
<poz-parbox> $\Longrightarrow$ [t ‖ b ‖ c]
<poz-parbox> $\Longrightarrow$ ε
<body> $\Longrightarrow$ `\mbox` <popis>
<body> $\Longrightarrow$ `\rule` <n-prumer><prumer><prumer> <body>
<body> $\Longrightarrow$ `\raisebox` <prumer><n-prumer><n-prumer><popis> <body>
<body> $\Longrightarrow$ `\savebox` {}<><><>{}
<body> $\Longrightarrow$ `\sbox` {}{}
<body> $\Longrightarrow$ `\newsavebox` {}
<body> $\Longrightarrow$ `\usebox` {}
<body> $\Longrightarrow$ `\fbox` <popis>
<body> $\Longrightarrow$ `\framebox` <n-prumer><poz-box><popis>
<body> $\Longrightarrow$ `\addcontentsline` {<text>}{<text>}{<text>} <body>
<body> $\Longrightarrow$ `\addcontents` {<text>}{<text>} <body>
<body> $\Longrightarrow$ `\listoffigures` <body>
<body> $\Longrightarrow$ `\listoftables` <body>
<body> $\Longrightarrow$ `\numberline` <pocet><popis> <body>
<body> $\Longrightarrow$ `\tableofcontents` <body>
<body> $\Longrightarrow$ `\cite` <cite>
<cite> $\Longrightarrow$ {text}

<body> $\Longrightarrow$ \appendix <body>

<text> $\Longrightarrow$ a-z0-9<text>
<text> $\Longrightarrow$ ε

<body> $\Longrightarrow$ \hypnetation<hypne> <body>
<hypne> $\Longrightarrow$ {text<dalsi-text>}
<dalsi-text> $\Longrightarrow$ text <dalsi-text>
<dalsi-text> $\Longrightarrow$ ε
<body> $\Longrightarrow$ \righthyphenmin=cislo <body>
<body> $\Longrightarrow$ \lefthyphenmin=cislo <body>
<body> $\Longrightarrow$ <spec-znak> <body>

<body> $\Longrightarrow$ \begin{multicols}<pocet> <body> \end{multicols}
<body> $\Longrightarrow$ \begin{center} <body> \end{center} <body>
<body> $\Longrightarrow$ \begin{flushleft} <body> \end{flushleft} <body>
<body> $\Longrightarrow$ \begin{flushright} <body> \end{flushright} <body>
<body> $\Longrightarrow$ \begin{description} <description> \end{description} <body>
<description> $\Longrightarrow$ \item<tucne> <body> <description>
<description> $\Longrightarrow$ ε
<tucne> $\Longrightarrow$ [<body>]
<tucne> $\Longrightarrow$ ε

<body> $\Longrightarrow$ \begin{enumerate} <enumerate> \end{enumerate} <body>
<enumerate> $\Longrightarrow$ \item <number> <body> <enumerate>
<enumerate> $\Longrightarrow$ ε
<number> $\Longrightarrow$ [cislo]
<number> $\Longrightarrow$ ε

<body> $\Longrightarrow$ \begin{math} <math> \end{math} <body>
<body> $\Longrightarrow$ (<math>) <body>
<body> $\Longrightarrow$ $ <math> $ <body>

<body> $\Longrightarrow$ \begin{displaymath} <math> \end{displaymath} <body>
<body> $\Longrightarrow$ [<math>] <body>
<body> $\Longrightarrow$ $$ <math> $$ <body>

<body> $\Longrightarrow$ \begin{equation} <mat-text> \end{equation} <body>
<body> $\Longrightarrow$ \begin{eqnarray<hvezda>} <eqnarray> \end{eqnarray} <body>

<body> $\Longrightarrow$ \begin{minipage}<placement><prumer> <body> \end{minipage} <body>

<body> $\Longrightarrow$ \begin{thebibliography}<popis> <thebibliography> \end{thebibliography}
<body>
<thebibliography> $\Longrightarrow$ \bibitem <nepovinne><popis> <thebibliography>
<thebibliography> $\Longrightarrow$ ε

<body> $\Longrightarrow$ \begin{verbatim<hvezda>} <body> \end{verbatim} <body>

<body> $\Longrightarrow$ \begin{alltt} <body> \end{alltt} <body>

<body> $\Longrightarrow$ \begin{verse} <verse> \end{verse} <body>
<verse> $\Longrightarrow$ <body> \\ <verse>
<verse> $\Longrightarrow \varepsilon$

<body> $\Longrightarrow$ \begin{itemize} <itemize> \end{itemize} <body>
<itemize> $\Longrightarrow$ \item <odrazka> <body>
<itemize> $\Longrightarrow \varepsilon$
<odrazka> $\Longrightarrow$ [znak]
<odrazka> $\Longrightarrow \varepsilon$

<body> $\Longrightarrow$ \begin{figure<hvezda>} <placement> <figure> \end{figure} <body>
<body> $\Longrightarrow$ \begin{picture}<rozmery><souradnice> <picture> \end{picture} <body>
<body> $\Longrightarrow$ \begin{list} <label> <spacing> <list> \end{list} <body>
<body> $\Longrightarrow$ \begin{quote} <body> \end{quote} <body>
<body> $\Longrightarrow$ \begin{quotation} <body> \end{quotation} <body>

<body> $\Longrightarrow$ \begin{tabbing} <tabbing> \end{tabbing} <body>
<body> $\Longrightarrow$ \begin{tabular}<placement><sloupce> <tabular> \end{tabular}
<body> $\Longrightarrow$ \begin{table<hvezda>}<placement> <table> \end{table} <body>
<body> $\Longrightarrow \varepsilon$

Kapitola 5

Interpretace

V této kapitole již je možno zúročit znalosti uvedené v předchozích kapitolách. Protože je ale interpretace rozsáhlým úkolem, jsou návrhy interpretačních pravidel rozděleny na ty, které jsou aktuálně použity v této verzi interpreteru a na ty, které využity zatím nejsou.

Využitá interpretační pravidla jsou uvedena v této kapitole, nevyužitá pravidla jsou uvedena v dodatcích.

HTML umožňuje rozsáhlé možnosti úprav svých elementů, ale nejsou ani zdaleka tak rozsáhlé jako u systému LaTeX. Proto jsou i intepretované pravidla pouze přibližným vzhledem LaTeXové předlohy.

5.1 Příkazy

Popisovat interpretaci jednotlivých příkazů by bylo velmi rozsáhlé, a ve srovnání s ostatními částmi diplomové práce by bylo ve velkém nepoměru. Navíc popis příkazů by mohl být nepřehledný, zvláště co se týká příkazů nastavení registrů a dalších pomocných, které nejsou interpretovány do HTML, ale řešeny pouze vnitřně v programu interpreteru.

V dodatku B jsou uvedeny všechny příkazy LaTeXu, které by měl program umět správně interpretovat. U pravidel není uveden návrh interpretace, celý výpis by to jen znepřehlednilo a navíc jak bylo zmíněno, ne všechny příkazy jsou řešeny triviálně a zbytečně by se podrobně rozebírala vnitřní funkcionalita programu.

5.2 Prostředí

`\begin{array}{sloupce}`
Prostředí pro tvorbu tabulek (matic) v matematickém režimu. Musí být uvozeno některým z matematických prostředí. Sloupce jsou definovány v hlavičce jako `{c|l|r}`. Takto jsme nadefinovali 3 sloupce, první z nich bude zarovnán na střed, druhý doleva a třetí doprava. Kolik písmen, tolik sloupců, znak — naznačuje, že má být vytvořena svislá čára. Hodnoty mezi jednotlivými sloupci jsou odděleny znakem &, řádky jsou odděleny znakem `\\`.

Interpretace tohoto prostředí proběhne převodem prostředí do obrázku gif a následným vložením tohoto obrázku do HTML kódu. Některé jednodušší rovnice by bylo možné interpretovat bez nutnosti převodu do gif obrázku, prozatím ale jsou takto převáděny všechny rovnice z prostředí `array`.

`\begin{center|flushleft|flushright}`
Prostředí, které definuje zarovnání bloku textu ohraničeném tímto prostředím na střed

(center), doleva (flushleft) nebo doprava (flushright).

Toto prostředí je možno interpetovat pomocí elementu `<div>` a jeho atributu `align`, který může nabývat hodnot `center`, `left` a `right`. Tím dojde k zarovnání bloku textu na požadovanou pozici. Ukončení prostředí je v HTML reprezentováno ukončujícím elementem `</div>`.

`\begin{description}`
Prostředí pro výčet popisovaných prostředí. Hesla jsou volitelným parametrem položka příkazu `\item`, která jsou automaticky vysázena tučně. Následuje i víceslovný popis položky. Pokud je potřeba vysázet uvnitř hesla hranatou závorku, musí být celé heslo uzavřeno do složených závorek, např. `\item[{položka}]`.

Pro interpretaci tohoto prostředí může posloužit HTML element pro definiční výčet `<dl>`. Ten dále obsahuje element `<dt>` označující výrazu, který bude definován a element `<dd>` označující definici předešlého výrazu. Pro lepší ilustraci je uveden příklad:

```
<dl>
  <dt>
    HTML
  </dt>
  <dd>
    HyperText Markup Language, ...
  </dd>
</dl>
```

Prostředí takto interpretované ale neodpovídá přesně své předloze, HTML chápe definiční výčet poněkud jinak než jak tomu je u LaTeXu.

`\begin{enumerate}`
Toto prostředí vytváří číslovaný seznam. Čísla se přiřazují automaticky. Pokud číslování chceme potlačit, použijeme volitelný parametr příkazu `\item[číslování]`, které danou položku očísluje podle parametru číslování. Jednotlivé položky jsou uvedeny na vlastním řádku.

Toto prostředí je opět relativně snadno interpretovatelné přímo pomocí HTML. HTML obsahuje element pro číslovaný seznam `<ol>`. Jednotlivé položky seznamu jsou označeny elementem `<li>`. Opět je uveden příklad:

```
<ol>
  <li>
    První položka
  </li>
  <li>
    Druhá položka
  </li>
</ol>
```

Oproti LaTeXu ale má HTML jen omezené možnosti, jak definovat způsob výpisu číslovaného seznamu.

`\begin{eqnarray}`
Sazba posloupnosti rovnic, které mají být vzájemně zarovnány. Celek se dělí na 3 části. Levou, střední a pravou. Celek je poté zarovnán tak, že střední část je zarovnána pod sebou

na střed, levé části na prapor vlevo, pravé na prapor vpravo. Střední část je ohraničena znak & střední část &, přechod na nový řádek je pak označen znakem \\.

Toto matematické prostředí je opět převáděno do gif formátu pomocí skriptu `textogif`. V dalších verzích programu by mohly být jednodušší rovnice zobrazovány pomocí HTML.

`\begin{equation}`
Prostředí pro vysazené matematické vztahy s automatickým číslováním. Číslo vzorce je umístěno na pravý okraj stránky v kulatých závorkách. Váže se k němu čítač equation.

Stejně jako předchozí je toto matematické prostředí opět převáděno do gif formátu pomocí skriptu `textogif`. V dalších verzích programu by mohly být jednodušší rovnice zobrazovány pomocí HTML.

`\begin{figure}[umístění]`
Plovoucí prostředí pro umístění obrázku na požadovanou pozici (kombinace až 4 písmen). H (here) vysází do místa, kde je umístěn jeho zdrojový kód, t (top), na horní okraj stránky, b (bottom), na dolní okraj stránky, p (page) vysází na samostatnou stránku, kde není běžný text, ale pouze plovoucí objekty. Složením těchto písmen sestavíme požadovanou prioritu umístění. Poté obvykle následuje vložení obrázku pomocí `\includegraphics` anebo prostředí `\picture`, které dále definuje vzhled obrázku (viz. níže). Příkaz `\caption{název}` definuje popisek, který bude u obrázku uveden s uvedením nápisu figure a pořadové číslo obrázku.

Prostředí figure by mohlo být interpretováné přímo do HTML, stačilo by k tomu jen to, aby byl vložený ten samý obrázek, který je vkládaný do tex kódu. Prostředí `picture`, které v prostředí `figure` může být vložené, ale není možné lehce interpretovat. V této fázi tedy je všechno převáděno pomocí skriptu do formátu gif a poté vloženo do HTML kódu. Později by mohlo být rozlišeno vkládání obrázku, který by se nepřeváděl a tvorba grafiky, která by i nadále využívala skript `textogif`.

`\begin{itemize}`
Prostředí vytváří formátované položky výčtu, standardně označené kuličkou (•). Každá položka je uvozena příkazem `\item`. Vnořené výčty jsou obecně posunuty o definovanou délku, lze to obejít např. nastavením hodnoty registru `\leftmargin`, tyto úpravy ale budu ignorovat. Položky výčtu první úrovně jsou označeny kuličkou (•), druhá úroveň je označena půlčtverčíkovou pomlčkou (-), třetí úroveň hvězdičkou (∗) a čtvrtá úroveň centrovanou tečkou (·).

Podobně jako prostředí `enumerate` je poměrně snadno interpretovatelné pomocí HTML. HTML obsahuje element pro nečíslovaný seznam `<ul>`. Jednotlivé položky seznamu jsou označeny elementem `<li>` Opět je uveden příklad:

```
<ul>
  <li>
    První položka
  </li>
  <li>
    Druhá položka
  </li>
</ul>
```

Zde jsou omezení HTML ještě markantnější než v případě interpretace prostředí `enumerate`. V HTML je možno nastavit odrážku jen z několika málo předdefinovaných možností, kdežto LaTeX má prakticky neomezené možnosti.

`\begin{list}{label}{spacing}`
Výčtové prostředí. Na rozdíl od itemize a podobných ale máme možnost formátovat odstavce položek tohoto výčtu. První povinný argument label definuje jakým návěštím bude označena položka, u jejíhož příkazu `\item` není uveden volitelný parametr, druhý, spacing, obsahuje nastavení formátovacích parametrů (`\topsep`, `\parskip`, `\partosep`, `\parsep`, atd.). Spojení uživatelských čítačů lze v druhém argumentu navázat pomocí `\usecounter{citac}`.

Toto prostředí by teoreticky bylo interpretovatelné, ale jeho rozsáhlé možnosti by tím nepochybně utrpěly. Proto prozatím bude převáděno do gif formátu.

`\begin{minipage}[pozice]{šířka}`
Prostředí chovající se jako samostatná malá stránka uvnitř jiné stránky. Definuje nepovinně pozici umístění stránky vzhledem k okolnímu textu, volitelný parametr určující výšku prostředí, volitelný parametr určující umístění obsahu a povinný parametr udávající šířku.

Z důvodu nedostatku času pro rozvedení problematiky prostředí minipage je toto zatím převáděno do gif formátu. Později by ale mohlo být interpetováno do HTML s použitím kaskádových stylů.

`\begin{picture}(šířka,výška)(x_souřadnice,y_souřadnice)`
Kreslení čar, přímek, šipek, libovolných textů. Uvnitř tohoto prostředí se mohou objevit pouze příkazy pro vložení obrazového objektu (`\put`, `\multiput`) a příkazy pro nastavení velikosti a typu písma nebo délek. Např. u příkazu `\begin{picture}(50,␣30)(10,␣20)` obrázek bude široký 50 jednotek, vysoký 30 jednotek a jeho levý dolní roh se nachází v souřadnicích [10, 20]. Souřadnice uvnitř prostředí se vztahují k bodu [0, 0], tomto případě mimo kreslící plochu. V tomto prostředí lze vložit různé frameboxy, čáry, ovály, kružnice, jejich nastavení atd.

Jak už bylo zmíněno u prostředí `figure`, je toto prostředí jen těžko interpretovatelné a proto je celé převáděno pomocí skriptu `textogif`.

`\begin{quotation|quote}`
Prostředí pro vysázení citátu, kde šířka odstavce je zmenšena o nastavené okraje o velikosti 1,5 em. V prostředí `quote` nemají odstavce zarážku a je mezi nimi nenulové odsazení. V prostředí `quotation` však jednotlivé odstavce zarážky mají a nemají naopak odsazení.

Pro interpretaci těchto dvou prostředí existuje v HTML jeden element, `<q>`. Jediný rozdíl v použití je nutnost potlačení odsazení odstavce tam, kde je interpretováno prostředí `quote`, což se děje využitím kaskádových stylů.

`\begin{tabbing}`
Prostředí umožňuje rozmístit části textu ve vodorovném směru podobně jako u psacího stroje pomocí tabelačních zarážek.

Toto prostředí je kompletně převáděno do gif formátu. Je velmi složité jej efektivně interpretovat do HTML.

`\begin{table}[umístění]`
Plovoucí prostředí pro umístění tabulky, umístění označuje způsob umístění tabulky na požadovanou pozici (kombinace až 4 písmen). H (here) sází přednostně do místa, kde je umístěn jeho zdrojový kód, t (top) sází přednostně na horní okraj stránky, b (bottom) na dolní okraj stránky, p (page) na samostatnou stránku, kde není běžný text, ale pouze plovoucí objekty. Složením těchto písmen sestavíme požadovanou prioritu umístění. Poté obvykle následuje prostředí tabular, které dále definuje vzhled tabulky (viz níže). Příkaz `\caption{název}` definuje popisek, který bude u tabulky uveden spolu s nápisem table a pořadovým číslem tabulky.

Pro sazbu tohoto plovoucího prostředí platí podobně jako u `figure`, že je celé převedeno do gif obrázku. Níže jsou uvedeny obě varianty prostředí `tabular`, které se mohou objevit samostatně anebo jako součást prostředí `table`. A stejně jako toto prostředí pro ně platí převod do gif obrázku.

`\begin{tabular}[pozice]{sloupce}`
Sazba textu a číslic přehledně upravená linkami. Nepovinný parametr pozice označuje, jakým způsobem dojde k napojení tabulky k okolnímu textu : t (top) napojení horním okrajem, nebo b (bottom) napojení dolním okrajem. Není-li parametr uveden, je tabulka připojena středem. Sloupce jsou definovány písmeny, označujícími způsob zarovnání textu v buňkách l(left) zarovnání doleva, r (right) doprava, c (center) doprostřed, p sazba textu položky do bloku o šířce, která je zadána ve svorce následující za p.

Příklad: `\begin{tabular}{|*2r*{15}{c|}*7{l|}}`: první dva sloupce obsahují text zarovnaný na pravý okraj, následuje 15 sloupců na střed se svislou čarou a za nimi 7 sloupců se zarovnáním na levý okraj. Jednotlivé položky tabulky jsou v řádku od sebe odděleny znakem &, na konci každého řádku je příkaz \\. Konec řádku tabulky jde přikázat i příkazem `\tabularnewline[míra]` míra udává svislou vzdálenost od následujícího řádku. Vodorovné čáry se nařizují na konci řádku příkazem `\hline`, pokud čára nemá vést přes celou šířku tabulky, použijeme příkaz `\cline{x␣-␣y}`, x a y jsou pořadová čísla sloupců odkud kam má vést vodorovná čára. Text přes více sloupců lze napsat pomocí příkazu `\multicolumn`. Příkazy jsou podrobněji rozebrány v dodatku B.

`\begin{tabular*}{délka}[pozice]{sloupce}`
Od předchozího prostředí se liší tím, že je možno vytvořit tabulku, jejíž celková šířka je zadána povinným parametrem délka. Jinak je šířka odvozena od obsahu sloupců.

`\begin{thebibliography}{nejdelší␣název}`
Prostředí pro tvorbu odkazů v textu, k jednoznačnému určení publikovaného díla. Už v průběhu celého textu jsou rozmístěny příkazy `\cite` nebo `\nocite`, které určují tyto odkazy do restříku referencí `\bibitem[název]{klíč}`

Povinným parametrem je návěští odkazu, použité v příkazu `\cite`. Dvojice `\bibitem` a `\cite` vytváří křížový odkaz. `\bibitem` automaticky produkuje pořadové číslo literatury. V tomto případě mají odkazy tvar pořadového čísla. Pokud chceme, aby odkaz vypadal jinak, použijeme nepovinný parametr název. `\cite[text]{klíče}` vytváří v textu odkaz do referencí literatury, podle definovaného klíče, definovaném v `\bibitem`. Text přidává ještě poznámku k odkazu. `\nocite{klíče}`.

V prostředí `thebibliography` dojde k vysázení seznamu použité literatury, což je možné provést taktéž pomocí HTML při dodržení daného formátování. To ale není nijak složité, a proto je intepretace tohoto prostředí poměrně jednoduchá.

`\begin{theindex}`
Příkazem `\index` v textu označíme slovo, nebo slova, která chceme zahrnout do indexu (slovo se nevypíše). Příkazem `\makeindex` aktivuje všechny příkazy `\index` a ty se vypíší do souboru .idx. Redefinicí prostředí theindex můžeme vypisovat rejstřík v různých tvarech.

Toto prostředí je prozatím zcela ignorováno, v dalších verzích se ale může objevit již funční.

`\begin{titlepage}`
Prostředí generující titulní stránku podle přání uživatele. Stránka je generována bez čísla stránky, stránka následující je očíslována číslem 1.

Sazba titulní stránky je snadnou záležitostí, vše záleží na dodržení konvencí sazby titulní stránky.

`\begin{verbatim}`
Text, který se nachází v prostředí verbatim, nepodléhá žádnému formátování, je vysázen tak, jak je napsán ve zdrojovém kódu. Standardně je nastaveno strojopisné písmo (*typewriter*).

Kratší úseky textu mohou být uvozeny příkazy `\verb` anebo `\verb*`, za oběma příkazy následuje hraniční znak, který označuje i konec textu který má být takto vypsán. Verze s hvězdičkou se liší pouze tím, že všechny mezery jsou sázeny přesně tak, jak jsou napsány. Jinak by při napsání více mezer byla vypsána pouze jedna.

V HTML je pro tyto účely používán element `<code>`, která nastaví font písma na zmiňovaný *typewriter*. Samotné dodržení definice, kdy mají být sázeny všechny příkazy LaTeXu, už závisí na programátorovi.

`\begin{verse}`
Prostředí pro sazbu veršů. Každá strofa začíná po prázdném řádku (kromě první). Jednotlivé verše jsou ukončeny příkazem `\\`.

V HTML bude toto prostředí sázeno na střed, při dosažení znaku `\\` pak bude ukončen řádek.

`\begin{math}`
Prostředí pro sazbu matematických vztahů uvnitř odstavce běžného textu. Pro lepší přehlednost lze ale použít i hraniční znaky `\(` matematický text `\)` anebo prostými znaky $.

`\begin{displaymath}`
Prostředí pro sazbu vysazených (zvýrazněných do vlastního odstavce, zarovnáno na střed) matematických vztahů. Podobně jako u prostředí math lze použít `\[` a `\]` anebo znaky $$.

Sazba obou prostředí, `math` i `displaymath`, bude provedena pomocí skriptu `textogif` a jejich vložení do textu či do samotného odstavce již pak určuje samotné prostředí.

5.3 Znaky a symboly

Některé znaky obsažené v systému LaTeX jsou interpretovatelné do HTML i bez využití skriptu `textogif`. Tyto jsou uvedeny v tabulce, jež následuje.Jsou uvedeny příkazy LaTeXu, jejich HTML ekvivalenty, i výsledný vysázený znak, či symbol.

Nejsou zde ale všechny znaky, které HTML umí zobrazit. Je to z důvodu, že matematické znaky, symboly a další speciální symboly jsou v první verzi programu převáděna přímo pomocí skriptu `textogif`. Znaky zobrazitelné pomocí HTML, ale prozatím nevyužity v tomto projektu jsou uvedeny v dodatku C.

Znaky a symboly, 1. část			
LaTeX příkaz symbolu	**vzhled symbolu**	**ekvivalent HTML**	**poznámka**
-	-	–	
--	–	—	
---	—	—	
\%	%	%	začátek komentáře
\{	{	{	
\}	}	}	
\uv{text}	“text”	„text”	
\textquotedblleft	“	“	
\textquotedblright	”	”	
\frqq	»	»	
\flqq	«	«	
\textquoteleft	‘	‘	
\textquoteright	’	’	
\quotesinglbase	‚	‚	
\S	§	§	
\&	&	&	
\dots	…	…	
\$	$	$	
\#	#	#	
\dag	†	†	
\ddag	‡	‡	
\P	¶	¶	
\copyright	©	©	
\pounds	£	£	
\’{x}	x́	&xgrave;	
\‘{x}	x̀	&xacute;	
\^{x}	x̂	◯	
\"{x}	ẍ	&xuml;	
\~{x}	x̃	&xtilde;	
\={x}	x̄	&xmacr;	
\.{x}	ẋ	&xmiddot;	
\u{x}	x̆	x	HTML neumí
\H{x}	x̋	&xuml;	
\t{xx}	x͡x	x	HTML neumí
\r{x}	x̊	&xring;	
\c{x}	x̧	&xcedil;	
\d{x}	x̣	x	HTML neumí
\b{x}	x̲	x	HTML neumí
\i	ı	i	HTML neumí
\j	ȷ	j	HTML neumí
\l	ł	ł	
\L	Ł	Ł	

Tabulka 5.1: Intepretované znaky, první část

Znaky a symboly, 2. část			
LaTeX příkaz symbolu	**vzhled symbolu**	**ekvivalent HTML**	**poznámka**
`\/`	xxxxxx	`&zwnj;`	mezera kurzivní korekce
`\,`	xxx xxx	` `	mezera 1/6 čtveříku
`\@`	xxxxxx	`&aelig;`	jiná mezera za tečkou
`~`	xxx xxx	` `	nezalomitelná mezera
`\␣`	xxx xxx	` `	mezislovní mezera
`\oe`	œ	`&oelig;`	
`\OE`	Œ	`&OElig;`	
`\ae`	æ	`&aelig;`	
`\AE`	Æ	`&AElig;`	
`\aa`	å	`&aring;`	
`\AA`	Å	`&Aring;`	
`\o`	ø	`&oslash;`	
`\O`	Ø	`&Oslash;`	
`\ss`	ß	`&szlig;`	
`?‘`	¿	`&iquest;`	
`!‘`	¡	`&iexcl;`	

Tabulka 5.2: Intepretované znaky, druhá část

Kapitola 6

Řešení

V této kapitole bude popsán výsledný program. Důraz je kladen na jeho instalaci, protože ke svému fungování potřebuje několik dalších programů, a na konfiguraci pomocných programů. Také je popsáno jeho používání a je shrnuta funkčnost programu včetně jeho funkčních omezení.

6.1 Popis programu

Běh programu probíhá ve dvou iteracích. První iterace slouží pro naplnění datových struktur, které plní funci obsahu, seznamu tabulek a obrázků a křížových odkazů v textu. Program v první iteraci neprovádí žádnou syntaktickou analýzu, pouze hlídá vstupní lexemy a pokud narazí při svém běhu na některý lexem, který se týká tvorby obsahu či křížového odkazu v textu (`\chapter{...}`, `\label{...}`, u prostředí `table` a `figure` také příkaz `\caption{...}`), uloží jej do této datové struktury. Přiřadí mu také správné pořadové číslo v rámci dané kapitoly a pokračuje dál ve vyhledávání. Níže je uveden popis této datové struktury (uvedeno v programovacím jazyce C):

```
typedef struct{
    char *nepovinny;            // uklada nepovinny parametr prikazu
    char *povinny;              // uklada povinny paramatr prikazu
    int cislovani;              // uklada aktualni poradi objektu v~kapitole
    int nadrizeny_part;         // uklada nadrizenou strukturu
    int nadrizeny_chapter;
    int nadrizeny_section;
    int nadrizeny_subsection;
    int nadrizeny_subsubsection;
    int nadrizeny_paragraph;
    int nadrizeny_subparagraph;
} cont_text;
```

Tato datová struktura je využita v další datové struktuře, která může dynamicky měnit počet jednotlivých svých částí. Je tak zaručen teoreticky nekonečný počet kapitol, popisků obrázků a dalších objektů. Objekty jako jsou `subsubsection` nebo `paragraph` nejsou pro potřeby sazby obsahu nutné, proto mají omezenou velikost a v první iteraci nejsou vůbec brány v potaz.

```
typedef struct{
```

```
    cont_text *part;
    cont_text *chapter;
    cont_text *section;
    cont_text *subsection;
    cont_text subsubsection[2];
    cont_text paragraph[2];
    cont_text subparagraph[2];
    cont_text *figure;
    cont_text *table;
} content;
```

Po skončení běhu první iterace je znovu otevřen soubor se vstupním zdrojovým kódem. Nyní již probíhá interpretace podle pravidel uvedených v kapitole 2.3. Pro detailní popis činnosti programu v druhé iteraci zde není dost prostoru, pro názornost je zde uvedena ilustrace průběhu interpretace, viz obrázek 6.1.

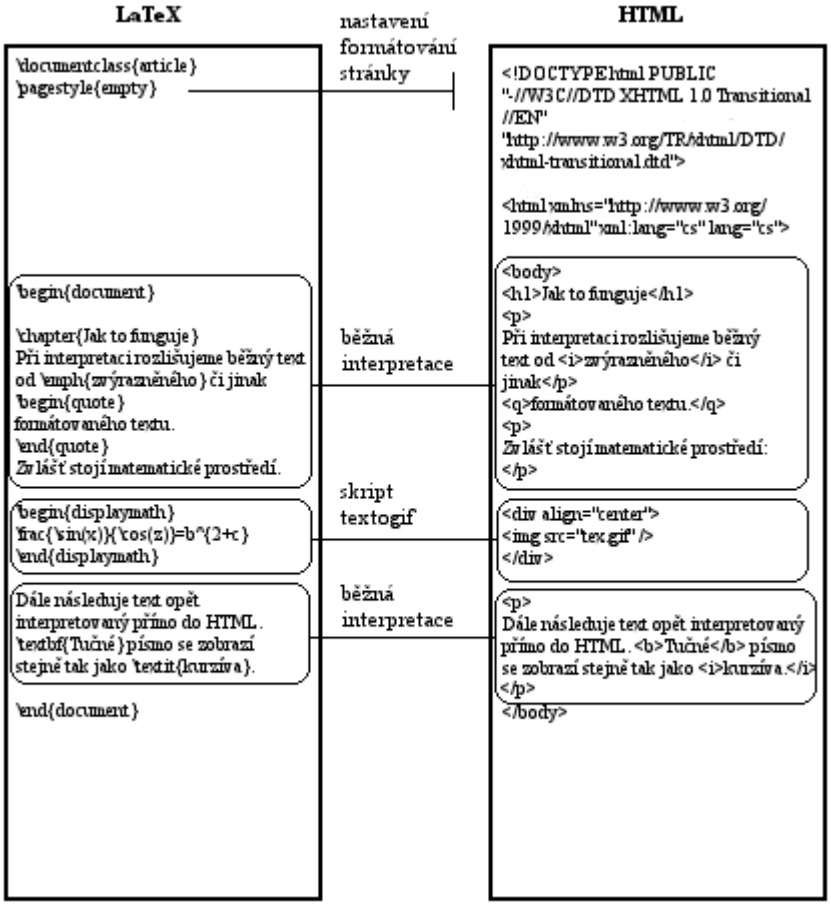
Pokud program při běhu narazí na příkaz pro vysázení obsahu, seznamu tabulek nebo seznamu obrázků (`\tableofcontent`, `\listoffigures` a `\listoftables`) jsou využita data z první iterace a jsou vysázena do podoby obsahu či zmiňovaného seznamu, včetně funkčních odkazů na odkazované kapitoly či obrázky.

Podobně se program zachová při sazbě křížových odkazů v textu, kdy vysází pořadové číslo daného odkazu, obrázku či tabulky.

Zotavení programu při přečtení chybného lexemu (vstupního příkazu) probíhá podobně jako u LaTeXu. Pokud je načten chybný lexem, interpret pokračuje v činnosti a snaží se interpretovat podle pravidel s tím, že chybné pravidlo je ignorováno. To může ale zapříčinit uváznutí v některém z prostředí a tím pádem neukončení programu. Bylo by dobré toto ošetřit a při přečtení uzavírací klauzule `\end{document}` násilně ukončit program, i když jeho činnost ještě neskončila.

Po úspěšném ukončení programu je výstupem adresář pojmenovaný podle vstupního souboru, který obsahuje HTML zdrojový kód internetových stránek, soubor s definicemi kaskádových stylů CSS, adresář s obrázky ve formátu gif, které byly vytvořeny skriptem `textogif` a adresář se zdrojovými kódy částí tex kódu, pomocí kterých byly vytvořeny gif obrázky.

Skript `textogif` totiž ke svému běhu potřebuje kompletní TeX zdrojový kód se všemi hlavičkami. Proto program, když narazí na začátek prostředí, které je převáděno do gif formátu, tento TeX kód ukládá do samostatného souboru a při volání skriptu `textogif` je použit. Výsledný obrázek je následně vložen do HTML kódu.



Obrázek 6.1: Ilustrace interpretace

6.2 Funkce a omezení programu

Samotný program provádí syntaktickou analýzu LaTeXu a zdrojový kód psaný v LaTeXu interpretuje do HTML kódu.

Program pracuje při druhé iteraci následujícím způsobem:

1. Na začátku analýzy program otevře zdrojový soubor, ve formátu `název_souboru.tex`
2. Proběhne syntaktická analýza hlavičky LaTeXového souboru. V něm jsou uvedeny atributy definující vzhled výsledného dokumentu. Tyto atributy se uloží do speciálního objektu. Ten bude určovat základní formát celého výsledného HTML dokumentu, stejně jako u LaTeXu.
3. V samotném těle zdrojového souboru LaTeXu probíhá syntaktická analýza a již analyzovaný text je zapisován podle definovaného formátování do HMTL souboru.
4. Pokud program narazí na začátek prostředí math, picture, tabular a dalších (více například v [3]), proběhne syntaktická analýza. Kód ale nebude interpretován, nýbrž převeden do samostatného LaTeXového souboru, který se odešle obslužnému skriptu `textogif`. Ten ke svému fungování potřebuje další programy, více viz [14]. Výsledný obrázek v gif formátu je vložen do HTML kódu. Vzorce, obrázky, tabulky a speciální znaky (například znaky řecké abecedy, matematické znaky) jsou tedy ve výsledku zobrazeny pomocí obrázků.
5. Po návratu z předchozích prostředí opět pokračuje klasická interpretace textu dokud není dosažen konec interpretovaného souboru.

Výsledný soubor bude uložen ve formátu HMTL na pevném disku uživatele v adresáři pojmenovaném podle vstupního souboru. Tento adresář je uložen ve stejném adresáři, kde je interpret.

Program výsledný dokument neformátuje jako běžný dokument na stránky, ale dokument je zde chápán jako celistvý blok textu s přesně umístěnými plovoucími objekty v tomto textu. Původně bylo zamýšleno text členit do bloků a ty graficky oddělit od sebe tak, aby byly patrné stránky, ale pro větší přehlednost výsledného dokumentu bylo od tohoto upuštěno.

Z toho ale plyne omezení správné interpretace prostředí `multicols`, které sází text do sloupců. Při absenci stránkování, ale nelze určit, kdy má nastat přechod do dalšího sloupce. Proto je toto prostředí, včetně příkazu `\twocolumn`, v podstatě ignorováno a interpretováno jako běžný text.

Další rozdíl oproti PDF předloze dokumentu je číslování stránek, které zde pochopitelně chybí. Proto například obsah neuvádí, na které straně se daná kapitola nachází, ale kliknutím na danou kapitolu je uživatel přesměrován do patřičné části dokumentu. Stejně tak není využíváno různých formátů číslování stránek a ani není sázena kapitola na novou stránku.

Kaskádové styly CSS jsou v této verzi programu využity spíše sporadicky a převážně při úpravě jednotlivých HTML elementů, například při zarámování textu. Stalo se tak hlavně z toho důvodu, že, jak už bylo zmíněno výše, je dokument brán jako celistvý blok textu. Původně bylo zamýšleno graficky oddělovat text do stránek, hlavně pomocí CSS. I přes tuto absenci jsou ale kaskádové styly využívány a našly zde své uplatnění.

HTML umožňuje nastavení různého obtékání textu kolem obrázků, ale program umisťuje obrázky jako samostatné odstavce, podobně jako LaTeX. Umožňuje zarovnání zobrazovaných obrázků k některému z okrajů, levému či pravému, či na střed.

Hlavní problém spočívá ve využívání mnohokrát zmiňovaného skriptu `textogif`, který se jeví jako vhodný pro převody malých obrázků, ale pro využití v šíři tohoto projektu poněkud zaostává. Pomalý převod obrázků zpomaluje celou interpretaci dokumentu a navíc nutnost ručně ukončovat konzoli GhostScriptu, která je volána tímto skriptem, vyžaduje uživatelovu přítomnost u počítače po celou dobu běhu programu. Některé obrázky takto generované dokonce nejde kvůli prozatím neznámé chybě využít, protože nejdou otevřít jakýmkoliv způsobem. Patrně je to způsobeno špatným konečným uložením obrázku.

Tento problém je patrně nejvhodněji řešitelný nahrazením tohoto skriptu vlastním programem, či modulem, který bude součástí interpreteru. Tento modul by poté sám obsluhoval převod TeX kódu do obrázků.

6.3 Instalace a používání programu

Na CD přiloženém k této diplomové práci jsou uloženy všechny potřebné programy pro běh programu. Pro správné fungování programu, hlavně pomocného skriptu `textogif`, který byl ručně upraven, je nutné nainstalovat další podpůrné programy přesně podle návodu. Další možností je upravit cesty k těmto programům přímo ve skriptu, samotný interpreter potřebuje, aby skript a zdrojové soubory určené k interpretaci byly uloženy ve stejném adresáři jako program.

Následuje výčet programů, které je nutno nainstalovat.

Perl - tento program je nutný pro interpretaci skriptu `textogif`, který je právě v Perlu naprogramován. Tento program může být nainstalován kamkoliv, záleží na přání uživatele.

LaTeX - na CD je přiložena odrůda LaTeXu, MikTeX, který se osvědčil jako spolehlivý s mnoha balíčky. Tento program musí být nainstalován do adresáře "C:\Miktex\"

GhostScript - slouží k interpretaci `ps` formátu. Většinou je už součástí systému LaTeX či některé jeho odrůdy. Adresář s GhostScriptem by měl být nainstalován do "C:\gs\"

NetPbm - toto je soubor programů provádějící nejrůznější operace s obrázky. Pomocí něj dochází k ořezání obrázků převáděných skriptem do gif formátu. NetPbm je potřeba nainstalovat do adresáře "C:\GnuWin32\"

textogif - tento skript zodpovědný za převod částí kódu do formátu gif je možné mít uložen kdekoliv, ovšem pod podmínkou, že ve stejném adresáři je i interpreter spolu s dokumentem určeným k interpretaci.

Není nezbytně nutné postupovat podle návodu a instalovat všechny programy na pevný disk `C:`, ale poté je potřeba upravit nastavení cest k programům definovaných ve skriptu `textogif`.

Samotný interpret se pouští buď z příkazového řádku příkazem

```
interpret.exe dokument.tex
```

kde `dokument.tex` může být libovolný název zdrojového TEX kódu. V případě, že dokument zadaného názvu nebyl nalezen, anebo uživatel neuvedl vůbec název dokumentu, je vyzván k zadání nového názvu. Program v této chvíli může ukončit zadáním příkazu `quit` a zmáčknutím tlačítka `Enter`. Podobně se postupuje, když uživatel pustí program přímo kliknutím myši na ikonku programu.

V případě úspěšného zadání názvu dokumentu se rozeběhne interpretace vstupního souboru.

Nakonec by bylo vhodné zmínit, že výsledek interpretace je uložen ve stejném adresáři jako je program spolu se skriptem a původním dokumentem. Zdrojové HTML kódy je pak možno umístit na internetový server a zpřístupnit je tak online.

Kapitola 7

Závěr

Tato diplomová práce naznačila skutečnost, že vytvoření interpreteru LaTeXu je sice velmi pracná, nikoliv však nereálná záležitost. Na 15 000 řádcích zdrojového kódu se podařilo správně interpretovat naprostou většinu LaTeXových příkazů, které uživatel může využít. Některé sice byly ignorovány, ale pro budoucí pokračování projektu jsou již předpřipraveny (např. práce s délkovými registry)

Výsledný program se zdá být rychlý a efektivní, jeho rychlost je ale do značné míry omezena během skriptu `textogif`, který se ukázal jako méně vhodná volba pomocného programu. Nehledě na skutečnost, že některé obrázky, vytvořené tímto skriptem, vůbec nejdou otevřít a tím pádem se ani nezobrazí v internetových stránkách.

Bohužel se nepodařilo zcela dodržet standardy W3C, avšak toto je pouze drobný problém, který je odstranitelný řádnou optimalizací kódu. Rovněž bylo upuštěno od zobrazování dokumentu po stránkách, stejně jak tomu je v běžném dokumentu. Toto rozhodnutí ale přispělo k větší celistvosti dokumentu a snad i k jeho přehlednosti. Stejně tak upuštění od objektového programování nehodnotím jako negativní věc, ale při důsledném programování je dopad jeho absence minimální.

Diplomová práce je nyní ve fázi, která umožňuje další vývoj tohoto systému a zlepšování jeho funkčnosti. Některé přípravné fáze již proběhly a tyto funkce jsou připraveny na použití v dalších verzích programu. V dalších verzích by mohlo být upuštěno od interpretace pomocí skriptu `textogif` a tento by mohl být nahrazen modulem, který by byl přímo součástí programu. Dále počet prostředí, interpretovaných pomocí tohoto skriptu by se měl snížit. Matematické prostředí by mohlo být vyhodnocováno a podle složitosti rovnic rozhodnuto, zda budou interpretovány pomocí prostředků HTML, anebo převedeny na obrázek. Prostředí `picture` a `figure` by vyhodnocovalo kdy je vkládán již hotový obrázek a ten by nebyl interpretován, ale rovnou vkládán do HTML kódu. A konečně prostředí `table` a `tabular` by mohlo být celé interpretováno pomocí prostředků implementovaných v HTML.

Jak je vidět, projekt je rozpracovaný a stále ve fázi vývoje a poskytuje velký prostor pro možná vylepšení a rozšíření. Mohu jen doufat, že program najde své uplatnění v běžném provozu a neskončí pouze jako jeden z mnoha neúspěšných projektů.

Literatura

[1] Castro, E.: HTML, XHTML a CSS, Brno, ComputerPress 2007, ISBN 978-80-251-1531-2

[2] Burget, R., Zeman, D.: Tvorba webových stránek, studijní opora, Brno, FIT VUT v Brně 2006

[3] Rybička, J.: LaTeXpro začátečníky, Brno, Konvoj 2003, ISBN 80-7302-049-1

[4] Lamport, L.: LaTeX: A document preparation system: User's quide and reference, Addison-Wesley Professional 1994, ISBN 0-201-52983-1

[5] Kopka, H., Daly, P.: A Guide to LaTeX 2_ε, Addison-Wesley Professional 2003, ISBN 0-321-17385-6

[6] Meduna, A.: Elements of Compiler Design, New York, Auerbach Publications 2008, ISBN 978-1-4200-6323-3

[7] Meduna, A.: Automata and Languages: Theory of Applications, London, Springer 2000, ISBN 1-85233-074-0

[8] Meduna, A., Lukáš, R.: Formal Languages and Compilers, studijní opora, Brno, FIT VUT v Brně 2004

[9] Hruška, T., Češka, M., Beneš, M.: Překladače, studijní opora, Brno, FIT VUT v Brně

[10] Češka, M., Rábová, Z.: Gramatiky a jazyky, studijní opora, Brno, FIT VUT v Brně

[11] Cascading Style Sheets [online]: dostupné na: http://www.css.org

[12] World Wide Web Consortium [online]: dostupné na: http://www.w3c.org

[13] DOC++ project [online]: dostupné na: http://docpp.sourceforge.net/

[14] Walker, J. [online]: dostupné na: http://www.fourmilab.ch/webtools/textogif

[15] TeX User Group [online]: dostupné na http://www.tug.org

[16] LaTeX project [online]: dostupné na http://www.latex-project.org

[17] Berners-Lee, T., Information Management: A Proposal, CERN 1990, [online]: dostupné na: http://www.w3.org/History/1989/proposal.html

[18] Berners-Lee, T., HTML Tags, [online]: dostupné na: http://www.w3.org/History/19921103-hypertext/hypertext/WWW/Markup/Tags.html

[19] Studentův wikiprůvodce životem na ČVUT, LL gramatiky [online]: dostupné na http://student.cvut.cz/cwut/index.php/LL_gramatiky

[20] Studentův wikiprůvodce životem na ČVUT, Syntaktická analýza metodou sdola nahoru [online]: dostupné na: http://student.cvut.cz/cwut/index.php/Syntaktick%C3%A1_anal%C3%BDza_metodou_zdola_nahoru

Seznam obrázků

Seznam tabulek

Dodatek A

Přepisovací pravidla

Z důvodu rozsáhlosti systému a časové tísni bylo rozhodnuto některé prostředí převádět skriptem textogif bez předchozí syntaktické analýzy. Stalo se tak u matematického prostředí, které stejně muselo být tímto způsobem interpretováno, ale například i u prostředí `tabular`, `table` a `tabbing`, které by sice bylo možno interpretovat pomocí HTML, ale kvůli jejich konečnému vzhledu a v neposlední řadě časové úspoře bylo rozhodnuto je interpretovat skriptem `textogif`.

Níže jsou ale uvedena nepoužitá pravidla, pro jejich možné budoucí využití.

<eqnarray> $\Longrightarrow$ <mat-text> & <mat-text> & <mat-text>
<eqnarray>
<eqnarray> $\Longrightarrow \varepsilon$
<mat-text> $\Longrightarrow$ `\setcounter` <popis><pocet> <mat-text>
<mat-text> $\Longrightarrow$ `\usecounter` <popis> <mat-text>
<mat-text> $\Longrightarrow$ <body> <mat-text>
<mat-text> $\Longrightarrow$ a-z0-9 <mat-text>
<mat-text> $\Longrightarrow$ `\!` <mat-text>
<mat-text> $\Longrightarrow$ `\;` <mat-text>
<mat-text> $\Longrightarrow$ `\␣` <mat-text>
<mat-text> $\Longrightarrow$ `\,` <mat-text>
<mat-text> $\Longrightarrow$ `\:` <mat-text>
<mat-text> $\Longrightarrow$ `\quad` <mat-text>
<mat-text> $\Longrightarrow$ `\qquad` <mat-text>
<mat-text> $\Longrightarrow$ `\overline` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\underline` <pom-mat><mat-text>
<pom-mat> $\Longrightarrow$ {<mat-text>}
<mat-text> $\Longrightarrow$ `^` <pom-mat>
<mat-text> $\Longrightarrow$ `^` <mat-text>
<mat-text> $\Longrightarrow$ `_` <pom-mat>
<mat-text> $\Longrightarrow$ `_` <mat-text>
<mat-text> $\Longrightarrow$ `\mathnormal` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\mathrm` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\mathbf` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\mathsf` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\mathit` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\mathtt` <pom-mat><mat-text>

<mat-text> $\Longrightarrow$ `\mathcal` <pom-mat><mat-text>
<mat-text> $\Longrightarrow$ `\displaystyle` <mat-text>
<mat-text> $\Longrightarrow$ `\textstyle` <mat-text>
<mat-text> $\Longrightarrow$ `\scriptstyle` <mat-text>
<mat-text> $\Longrightarrow$ `\scriptscriptstyle` <mat-text>
<mat-text> $\Longrightarrow$ `\alpha` <mat-text>
<mat-text> $\Longrightarrow$ `\beta` <mat-text>
<mat-text> $\Longrightarrow$ `\gamma` <mat-text>
<mat-text> $\Longrightarrow$ `\delta` <mat-text>
<mat-text> $\Longrightarrow$ `\epsilon` <mat-text>
<mat-text> $\Longrightarrow$ `\varepsilon` <mat-text>
<mat-text> $\Longrightarrow$ `\zeta` <mat-text>
<mat-text> $\Longrightarrow$ `\eta` <mat-text>
<mat-text> $\Longrightarrow$ `\theta` <mat-text>
<mat-text> $\Longrightarrow$ `\vartheta` <mat-text>
<mat-text> $\Longrightarrow$ `\iota` <mat-text>
<mat-text> $\Longrightarrow$ `\kappa` <mat-text>
<mat-text> $\Longrightarrow$ `\lambda` <mat-text>
<mat-text> $\Longrightarrow$ `\mu` <mat-text>
<mat-text> $\Longrightarrow$ `\nu` <mat-text>
<mat-text> $\Longrightarrow$ `\xi` <mat-text>
<mat-text> $\Longrightarrow$ `\o` <mat-text>
<mat-text> $\Longrightarrow$ `\pi` <mat-text>
<mat-text> $\Longrightarrow$ `\varpi` <mat-text>
<mat-text> $\Longrightarrow$ `\rho` <mat-text>
<mat-text> $\Longrightarrow$ `\varrho` <mat-text>
<mat-text> $\Longrightarrow$ `\sigma` <mat-text>
<mat-text> $\Longrightarrow$ `\varsigma` <mat-text>
<mat-text> $\Longrightarrow$ `\tau` <mat-text>
<mat-text> $\Longrightarrow$ `\upsilon` <mat-text>
<mat-text> $\Longrightarrow$ `\phi` <mat-text>
<mat-text> $\Longrightarrow$ `\varphi` <mat-text>
<mat-text> $\Longrightarrow$ `\chi` <mat-text>
<mat-text> $\Longrightarrow$ `\psi` <mat-text>
<mat-text> $\Longrightarrow$ `\omega` <mat-text>
<mat-text> $\Longrightarrow$ `\Gamma` <mat-text>
<mat-text> $\Longrightarrow$ `\Delta` <mat-text>
<mat-text> $\Longrightarrow$ `\Theta` <mat-text>
<mat-text> $\Longrightarrow$ `\Lambda` <mat-text>
<mat-text> $\Longrightarrow$ `\Xi` <mat-text>
<mat-text> $\Longrightarrow$ `\Pi` <mat-text>
<mat-text> $\Longrightarrow$ `\Sigma` <mat-text>
<mat-text> $\Longrightarrow$ `\Upsilon` <mat-text>
<mat-text> $\Longrightarrow$ `\Phi` <mat-text>
<mat-text> $\Longrightarrow$ `\Psi` <mat-text>
<mat-text> $\Longrightarrow$ `\Omega` <mat-text>
<mat-text> $\Longrightarrow$ `\leftarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Leftarrow` <mat-text>

<mat-text> $\Longrightarrow$ `\rightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Rightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\leftrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Leftrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\mapsto` <mat-text>
<mat-text> $\Longrightarrow$ `\hookleftarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\leftharpoonup` <mat-text>
<mat-text> $\Longrightarrow$ `\leftharpoondown` <mat-text>
<mat-text> $\Longrightarrow$ `\rightleftharpoons` <mat-text>
<mat-text> $\Longrightarrow$ `\longleftarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\longrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Longleftarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Longrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\longleftrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Longleftrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\longmapsto` <mat-text>
<mat-text> $\Longrightarrow$ `\hookrightarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\rightharpoonup` <mat-text>
<mat-text> $\Longrightarrow$ `\rightharpoondown` <mat-text>
<mat-text> $\Longrightarrow$ `\leadsto` <mat-text>
<mat-text> $\Longrightarrow$ `\uparrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Uparrow` <mat-text>
<mat-text> $\Longrightarrow$ `\downarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Downarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\updownarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\Updownarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\nearrow` <mat-text>
<mat-text> $\Longrightarrow$ `\searrow` <mat-text>
<mat-text> $\Longrightarrow$ `\swarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\nwarrow` <mat-text>
<mat-text> $\Longrightarrow$ `\leq` <mat-text>
<mat-text> $\Longrightarrow$ `\prec` <mat-text>
<mat-text> $\Longrightarrow$ `\preceq` <mat-text>
<mat-text> $\Longrightarrow$ `\ll` <mat-text>
<mat-text> $\Longrightarrow$ `\subset` <mat-text>
<mat-text> $\Longrightarrow$ `\sqsubset` <mat-text>
<mat-text> $\Longrightarrow$ `\sqsubseteq` <mat-text>
<mat-text> $\Longrightarrow$ `\in` <mat-text>
<mat-text> $\Longrightarrow$ `\vdash` <mat-text>
<mat-text> $\Longrightarrow$ `\geq` <mat-text>
<mat-text> $\Longrightarrow$ `\succ` <mat-text>
<mat-text> $\Longrightarrow$ `\succeq` <mat-text>
<mat-text> $\Longrightarrow$ `\gg` <mat-text>
<mat-text> $\Longrightarrow$ `\supset` <mat-text>
<mat-text> $\Longrightarrow$ `\supseteq` <mat-text>
<mat-text> $\Longrightarrow$ `\sqsupset` <mat-text>
<mat-text> $\Longrightarrow$ `\sqsupseteq` <mat-text>
<mat-text> $\Longrightarrow$ `\ni` <mat-text>

<mat-text> ⟹ \dashv <mat-text>
<mat-text> ⟹ \equiv <mat-text>
<mat-text> ⟹ \sim <mat-text>
<mat-text> ⟹ \simeq <mat-text>
<mat-text> ⟹ \asymp <mat-text>
<mat-text> ⟹ \approx <mat-text>
<mat-text> ⟹ \cong <mat-text>
<mat-text> ⟹ \neq <mat-text>
<mat-text> ⟹ \doteq <mat-text>
<mat-text> ⟹ \propto <mat-text>
<mat-text> ⟹ \models <mat-text>
<mat-text> ⟹ \perp <mat-text>
<mat-text> ⟹ \mid <mat-text>
<mat-text> ⟹ \parallel <mat-text>
<mat-text> ⟹ \bowtie <mat-text>
<mat-text> ⟹ \Join <mat-text>
<mat-text> ⟹ \smile <mat-text>
<mat-text> ⟹ \frown <mat-text>
<mat-text> ⟹ \pm <mat-text>
<mat-text> ⟹ \mp <mat-text>
<mat-text> ⟹ \times <mat-text>
<mat-text> ⟹ \div <mat-text>
<mat-text> ⟹ \ast <mat-text>
<mat-text> ⟹ \star <mat-text>
<mat-text> ⟹ \circ <mat-text>
<mat-text> ⟹ \bullet <mat-text>
<mat-text> ⟹ \cdot <mat-text>
<mat-text> ⟹ \cap <mat-text>
<mat-text> ⟹ \cup <mat-text>
<mat-text> ⟹ \uplus <mat-text>
<mat-text> ⟹ \sqcap <mat-text>
<mat-text> ⟹ \sqcup <mat-text>
<mat-text> ⟹ \vee <mat-text>
<mat-text> ⟹ \wedge <mat-text>
<mat-text> ⟹ \setminus <mat-text>
<mat-text> ⟹ \wr <mat-text>
<mat-text> ⟹ \diamond <mat-text>
<mat-text> ⟹ \bigtriangleup <mat-text>
<mat-text> ⟹ \bigtriangledown <mat-text>
<mat-text> ⟹ \triangleleft <mat-text>
<mat-text> ⟹ \triangleright <mat-text>
<mat-text> ⟹ \lhd <mat-text>
<mat-text> ⟹ \rhd <mat-text>
<mat-text> ⟹ \unlhd <mat-text>
<mat-text> ⟹ \unrhd <mat-text>
<mat-text> ⟹ \oplus <mat-text>
<mat-text> ⟹ \ominus <mat-text>
<mat-text> ⟹ \otimes <mat-text>

<mat-text> $\Longrightarrow$ `\oslash` <mat-text>
<mat-text> $\Longrightarrow$ `\odot` <mat-text>
<mat-text> $\Longrightarrow$ `\bigcirc` <mat-text>
<mat-text> $\Longrightarrow$ `\dagger` <mat-text>
<mat-text> $\Longrightarrow$ `\ddagger` <mat-text>
<mat-text> $\Longrightarrow$ `\amalg` <mat-text>
<mat-text> $\Longrightarrow$ `\aleph` <mat-text>
<mat-text> $\Longrightarrow$ `\hbar` <mat-text>
<mat-text> $\Longrightarrow$ `\imath` <mat-text>
<mat-text> $\Longrightarrow$ `\jmath` <mat-text>
<mat-text> $\Longrightarrow$ `\ell` <mat-text>
<mat-text> $\Longrightarrow$ `\wp` <mat-text>
<mat-text> $\Longrightarrow$ `\Re` <mat-text>
<mat-text> $\Longrightarrow$ `\Im` <mat-text>
<mat-text> $\Longrightarrow$ `\mho` <mat-text>
<mat-text> $\Longrightarrow$ `\prime` <mat-text>
<mat-text> $\Longrightarrow$ `\emptyset` <mat-text>
<mat-text> $\Longrightarrow$ `\nabla` <mat-text>
<mat-text> $\Longrightarrow$ `\surd` <mat-text>
<mat-text> $\Longrightarrow$ `\top` <mat-text>
<mat-text> $\Longrightarrow$ `\bot` <mat-text>
<mat-text> $\Longrightarrow$ `\$\|$` <mat-text>
<mat-text> $\Longrightarrow$ `\angle` <mat-text>
<mat-text> $\Longrightarrow$ `\forall` <mat-text>
<mat-text> $\Longrightarrow$ `\exists` <mat-text>
<mat-text> $\Longrightarrow$ `\neg` <mat-text>
<mat-text> $\Longrightarrow$ `\flat` <mat-text>
<mat-text> $\Longrightarrow$ `\natural` <mat-text>
<mat-text> $\Longrightarrow$ `\sharp` <mat-text>
<mat-text> $\Longrightarrow$ `\backslash` <mat-text>
<mat-text> $\Longrightarrow$ `\partial` <mat-text>
<mat-text> $\Longrightarrow$ `\infty` <mat-text>
<mat-text> $\Longrightarrow$ `\Box` <mat-text>
<mat-text> $\Longrightarrow$ `\Diamond` <mat-text>
<mat-text> $\Longrightarrow$ `\triangle` <mat-text>
<mat-text> $\Longrightarrow$ `\clubsuit` <mat-text>
<mat-text> $\Longrightarrow$ `\diamondsuit` <mat-text>
<mat-text> $\Longrightarrow$ `\heartsuit` <mat-text>
<mat-text> $\Longrightarrow$ `\spadesuit` <mat-text>
<mat-text> $\Longrightarrow$ `\sum` <mat-text>
<mat-text> $\Longrightarrow$ `\prod` <mat-text>
<mat-text> $\Longrightarrow$ `\coprod` <mat-text>
<mat-text> $\Longrightarrow$ `\int` <int-pom> <mat-text>
<mat-text> $\Longrightarrow$ `\oint` <int-pom> <mat-text>
<int-pom> $\Longrightarrow$ `\limits` <mat-text>
<int-pom> $\Longrightarrow \varepsilon$
<mat-text> $\Longrightarrow$ `\bigcap` <mat-text>
<mat-text> $\Longrightarrow$ `\bigcup` <mat-text>

<mat-text> $\Longrightarrow$ `\bigsqcup` <mat-text>
<mat-text> $\Longrightarrow$ `\bigvee` <mat-text>
<mat-text> $\Longrightarrow$ `\bigwedge` <mat-text>
<mat-text> $\Longrightarrow$ `\bigodot` <mat-text>
<mat-text> $\Longrightarrow$ `\bigotimes` <mat-text>
<mat-text> $\Longrightarrow$ `\bigoplus` <mat-text>
<mat-text> $\Longrightarrow$ `\biguplus` <mat-text>
<mat-text> $\Longrightarrow$ `\arcos` <mat-text>
<mat-text> $\Longrightarrow$ `\sinh` <mat-text>
<mat-text> $\Longrightarrow$ `\Pr` <mat-text>
<mat-text> $\Longrightarrow$ `\log` <mat-text>
<mat-text> $\Longrightarrow$ `\liminf` <mat-text>
<mat-text> $\Longrightarrow$ `\cos` <mat-text>
<mat-text> $\Longrightarrow$ `\arcsin` <mat-text>
<mat-text> $\Longrightarrow$ `\sup` <mat-text>
<mat-text> $\Longrightarrow$ `\sec` <mat-text>
<mat-text> $\Longrightarrow$ `\max` <mat-text>
<mat-text> $\Longrightarrow$ `\csc` <mat-text>
<mat-text> $\Longrightarrow$ `\cosh` <mat-text>
<mat-text> $\Longrightarrow$ `\arctan` <mat-text>
<mat-text> $\Longrightarrow$ `\tan` <mat-text>
<mat-text> $\Longrightarrow$ `\sin` <mat-text>
<mat-text> $\Longrightarrow$ `\epx` <mat-text>
<mat-text> $\Longrightarrow$ `\deg` <mat-text>
<mat-text> $\Longrightarrow$ `\cot` <mat-text>
<mat-text> $\Longrightarrow$ `\arg` <mat-text>
<mat-text> $\Longrightarrow$ `\tanh` <mat-text>
<mat-text> $\Longrightarrow$ `\ker` <mat-text>
<mat-text> $\Longrightarrow$ `\gcd` <mat-text>
<mat-text> $\Longrightarrow$ `\det` <mat-text>
<mat-text> $\Longrightarrow$ `\coth` <mat-text>
<mat-text> $\Longrightarrow$ `\limsup` <mat-text>
<mat-text> $\Longrightarrow$ `\lg` <mat-text>
<mat-text> $\Longrightarrow$ `\hom` <mat-text>
<mat-text> $\Longrightarrow$ `\dim` <mat-text>
<mat-text> $\Longrightarrow$ `\min` <mat-text>
<mat-text> $\Longrightarrow$ `\ln` <mat-text>
<mat-text> $\Longrightarrow$ `\lim` <mat-text>
<mat-text> $\Longrightarrow$ `\inf` <mat-text>
<mat-text> $\Longrightarrow$ (<mat-text>
<mat-text> $\Longrightarrow$) <mat-text>
<mat-text> $\Longrightarrow$ [<mat-text>
<mat-text> $\Longrightarrow$] <mat-text>
<mat-text> $\Longrightarrow$ `\{`<mat-text>
<mat-text> $\Longrightarrow$ `\}`<mat-text>
<mat-text> $\Longrightarrow$ `\lfloor` <mat-text>
<mat-text> $\Longrightarrow$ `\rfloor` <mat-text>
<mat-text> $\Longrightarrow$ `\lceil` <mat-text>

<mat-text> ⟹ `\rceil` <mat-text>
<mat-text> ⟹ `\langle` <mat-text>
<mat-text> ⟹ `\rangle` <mat-text>
<mat-text> ⟹ `/` <mat-text>
<mat-text> ⟹ `\|` <mat-text>
<mat-text> ⟹ `\bmod` <mat-text>
<mat-text> ⟹ `\pmod` <mat-text>
<mat-text> ⟹ `\overbrace` <mat-text>
<mat-text> ⟹ `\underbrace` <mat-text>
<mat-text> ⟹ `\frac` <pom-mat><pom-mat> <mat-text>
<mat-text> ⟹ `\sqrt` <N><pom-mat> <mat-text>
<mat-text> ⟹ `\left` <zavorky> <mat-text> `\right` <zavorky> <mat-text>
<zavorky> ⟹ (
<zavorky> ⟹)
<zavorky> ⟹ [
<zavorky> ⟹]
<zavorky> ⟹ `\{`
<zavorky> ⟹ `\}`
<zavorky> ⟹ `\lfloor`
<zavorky> ⟹ `\rfloor`
<zavorky> ⟹ `\lceil`
<zavorky> ⟹ `\rceil`
<zavorky> ⟹ `\langle`
<zavorky> ⟹ `\rangle`
<zavorky> ⟹ `\uparrow`
<zavorky> ⟹ `\Uparrow`
<zavorky> ⟹ `\downarrow`
<zavorky> ⟹ `\Downarrow`
<zavorky> ⟹ `\updownarrow`
<zavorky> ⟹ `\Updownarrow`
<zavorky> ⟹ `/`
<zavorky> ⟹ `\backslash`
<zavorky> ⟹ `$\|$`
<zavorky> ⟹ `\$\|$`
<zavorky> ⟹ .
<mat-text> ⟹ `\cdots` <mat-text>
<mat-text> ⟹ `\vdots` <mat-text>
<mat-text> ⟹ `\ddots` <mat-text>
<mat-text> ⟹ `\ldots` <mat-text>
<mat-text> ⟹ `\widehat` <pom-mat> <mat-text>
<mat-text> ⟹ `\widetilde` <pom-mat> <mat-text>
<mat-text> ⟹ `\vec` <pom-mat> <mat-text>
<mat-text> ⟹ `\over` <mat-text>
<mat-text> ⟹ `\atop` <mat-text>
<mat-text> ⟹ `\above` <mat-text>
<mat-text> ⟹ `\choose` <mat-text>
<mat-text> ⟹ `\brace` <mat-text>
<mat-text> ⟹ `\brack` <mat-text>

<mat-text> $\Longrightarrow$ `\stackrel`<pom-mat><pom-mat> <mat-text>
<mat-text> $\Longrightarrow$ `\hat` <pom-mat>
<mat-text> $\Longrightarrow$ `\check` <pom-mat>
<mat-text> $\Longrightarrow$ `\breve` <pom-mat>
<mat-text> $\Longrightarrow$ `\acute` <pom-mat>
<mat-text> $\Longrightarrow$ `\grave` <pom-mat>
<mat-text> $\Longrightarrow$ `\tilde` <pom-mat>
<mat-text> $\Longrightarrow$ `\bar` <pom-mat>
<mat-text> $\Longrightarrow$ `\dot` <pom-mat>
<mat-text> $\Longrightarrow$ `\ddot` <pom-mat>
<mat-text> $\Longrightarrow$ `\begin`{array}<array-sloupce> <array> `\end`{array} <mat-text>
<array-sloupce> $\Longrightarrow$ { kombinace l, c, r, $\|$ }
<array> $\Longrightarrow$ <mat-text> <pok-array>
<array> $\Longrightarrow$ ε
<pok-array> $\Longrightarrow$ & <array>
<pok-array> $\Longrightarrow$ `\\` <array>
<pok-array> $\Longrightarrow$ ε

<body> $\Longrightarrow$ `\begin`{figure<hvezda>} <placement> <figure> `\end`{figure} <body>
<placement> $\Longrightarrow$ [kombinace h,t,b,p]
<placement> $\Longrightarrow$ ε
<figure> $\Longrightarrow$ <prikaz> `\includegraphics` <cesta> <prikaz>
<figure> $\Longrightarrow$ <prikaz> `\begin`{picture}<rozmery><souradnice> <picture> `\end`{picture}
<prikaz>
<prikaz> $\Longrightarrow$ `\centering` <prikaz>
<prikaz> $\Longrightarrow$ `\label` <popis> <prikaz>
<prikaz> $\Longrightarrow$ `\caption` <nepovinne> <popis> <prikaz>
<prikaz> $\Longrightarrow$ ε
<popis> $\Longrightarrow$ {<body>}
<nepovinne> $\Longrightarrow$ [<body>]
<nepovinne> $\Longrightarrow$ ε

<body> $\Longrightarrow$ `\begin`{itemize} <itemize> `\end`{itemize} <body>
<itemize> $\Longrightarrow$ `\item` <odrazka> <body>
<itemize> $\Longrightarrow$ ε
<odrazka> $\Longrightarrow$ [znak]
<odrazka> $\Longrightarrow$ ε

<body> $\Longrightarrow$ `\begin`{list} <label> <spacing> <list> `\end`{list} <body>
<label> $\Longrightarrow$ {<in-label>}
<in-label> $\Longrightarrow$ <body> <in-label>
<in-label> $\Longrightarrow$ <prikaz-list> <in-label>
<in-label> $\Longrightarrow$ ε
<spacing> $\Longrightarrow$ {<prikaz-list>}
<prikaz-list> $\Longrightarrow$ `\usecounter` <popis> <prikaz-list>
<prikaz-list> $\Longrightarrow$ `\setlength` <what><prumer>
<what> $\Longrightarrow$ {`\labelsep`}
<what> $\Longrightarrow$ {`\labelwidth`}

<what> $\Longrightarrow$ {\topsep}
<what> $\Longrightarrow$ {\parskip}
<what> $\Longrightarrow$ {\partopsep}
<what> $\Longrightarrow$ {\rightmargin}
<what> $\Longrightarrow$ {\leftmargin}
<what> $\Longrightarrow$ {\itemindent}
<what> $\Longrightarrow$ {\listparindent}
<what> $\Longrightarrow$ {\parsep}
<what> $\Longrightarrow$ {\itemsep}
<body> $\Longrightarrow$ \begin{trivlist} <rovnani> <description> \end{trivlist}
<rovnani> $\Longrightarrow$ \centering
<rovnani> $\Longrightarrow$ \flushleft
<rovnani> $\Longrightarrow$ \flushright
<rovnani> $\Longrightarrow$ ε

<body> $\Longrightarrow$ \begin{picture}<rozmery><souradnice> <picture> \end{picture} <body>
<rozmery> $\Longrightarrow$ (cislo, cislo)
<souradnice> $\Longrightarrow$ (cislo, cislo)
<souradnice> $\Longrightarrow$ ε
<pocet> $\Longrightarrow$ {cele-cislo}
<procento> $\Longrightarrow$ {cislo}
<N> $\Longrightarrow$ [cele-cislo]
<prumer> $\Longrightarrow$ {<mira>}
<n-prumer> $\Longrightarrow$ [<mira>]
<n-prumer> $\Longrightarrow$ ε
<mira> $\Longrightarrow$ cislo pt
<mira> $\Longrightarrow$ cislo mm
<mira> $\Longrightarrow$ cislo cm
<mira> $\Longrightarrow$ cislo em
<mira> $\Longrightarrow$ cislo in
<mira> $\Longrightarrow$ cislo cc
<mira> $\Longrightarrow$ cislo dd
<mira> $\Longrightarrow$ cislo ex
<mira> $\Longrightarrow$ cislo pc
<mira> $\Longrightarrow$ cislo pt
<mira> $\Longrightarrow$ cislo sp
<mira> $\Longrightarrow$ cislo \linewidth
<picture> $\Longrightarrow$ \linethickness <prumer> <picture>
<picture> $\Longrightarrow$ \thicklines <picture>
<picture> $\Longrightarrow$ \thinlines <picture>
<picture> $\Longrightarrow$ \put <souradnice> <objekt> <picture>
<picture> $\Longrightarrow$ \multiput <rozmery> <rozmery> <pocet> <picture2>
<picture2> $\Longrightarrow$ {<picture>}
<picture> $\Longrightarrow$ \qbezier <N><rozmery><rozmery><rozmery>
<picture> $\Longrightarrow$ \bezier <pocet><rozmery><rozmery><rozmery>
<picture> $\Longrightarrow$ \savebox <picture><rozmery><poz><objekt>
<picture> $\Longrightarrow$ \scalebox <procento> <picture>
<picture> $\Longrightarrow$ \resizebox <prumer><prumer> <picture>

<picture> $\Longrightarrow$ `\rotatebox` <procento> <picture>
<picture> $\Longrightarrow$ `\reflectbox` <popis> <picture>
<picture> $\Longrightarrow \varepsilon$
<objekt> $\Longrightarrow$ {<objekt2>}
<objekt2> $\Longrightarrow$ `\framebox` <rozmery> <poz> <objekt>
<poz> $\Longrightarrow$ [kombinace dvou pismen z t, c, b, l, r]
<poz> $\Longrightarrow \varepsilon$
<objekt2> $\Longrightarrow$ <body> <objekt>
<objekt2> $\Longrightarrow$ `\dashbox` <prumer> <rozmery> <poz> <objekt>
<objekt2> $\Longrightarrow$ `\makebox` <rozmery> <poz> <objekt>
<objekt2> $\Longrightarrow$ `\put` <rozmery> <objekt>
<objekt2> $\Longrightarrow$ `\vector` <rozmery> <pocet>
<objekt2> $\Longrightarrow$ `\circle` <hvezda><prumer>
<objekt2> $\Longrightarrow$ `\oval` <rozmery><poz>
<objekt2> $\Longrightarrow$ `\line` <rozmery><pocet>
<objekt2> $\Longrightarrow$ `\shortstack` <zarovnani> <body>
<zarovnani> $\Longrightarrow$ [l $\|$ r]
<objekt2> $\Longrightarrow$ `\frame` <objekt>

<body> $\Longrightarrow$ `\begin`{quote} <body> `\end`{quote} <body>
<body> $\Longrightarrow$ `\begin`{quotation} <body> `\end`{quotation} <body>

<body> $\Longrightarrow$ `\begin`{tabbing} <tabbing> `\end`{tabbing} <body>
<tabbing> $\Longrightarrow$ <body> <tabbing>
<tabbing> $\Longrightarrow$ `\'` <tabbing>
<tabbing> $\Longrightarrow$ `\‘` <tabbing>
<tabbing> $\Longrightarrow$ `\+` <tabbing>
<tabbing> $\Longrightarrow$ `\-` <tabbing>
<tabbing> $\Longrightarrow$ `\` < <tabbing>
<tabbing> $\Longrightarrow$ `\=` <tabbing>
<tabbing> $\Longrightarrow$ `\` > <tabbing>
<tabbing> $\Longrightarrow$ `\a'`{znak} <tabbing>
<tabbing> $\Longrightarrow$ `\a‘`{znak} <tabbing>
<tabbing> $\Longrightarrow$ `\a=`{znak} <tabbing>
<tabbing> $\Longrightarrow$ `\kill` <tabbing>
<tabbing> $\Longrightarrow$ `\poptabs` <tabbing>
<tabbing> $\Longrightarrow$ `\pushtabs` <tabbing>
<tabbing> $\Longrightarrow$ `\tabbingsep` <tabbing>
<tabbing> $\Longrightarrow$ `\\` <tabbing>
<tabbing> $\Longrightarrow \varepsilon$

<body> $\Longrightarrow$ `\begin`{tabular}<placement><sloupce> <tabular> `\end`{tabular}
<body> $\Longrightarrow$ `\begin`{table<hvezda>}<placement> <table> `\end`{table} <body>
<table> $\Longrightarrow$ <prikaz> `\begin`{tabular}<placement><sloupce> <tabular> `\end`{tabular}
<prikaz>
<table> $\Longrightarrow$ <prikaz> `\begin`{tabular*}<prumer><placement><sloupce> <tabular>
`\end`{tabular} <prikaz>
<prikaz> $\Longrightarrow$ `\centering` <prikaz>

<prikaz> $\Longrightarrow$ `\flushleft` <prikaz>
<prikaz> $\Longrightarrow$ `\flushright` <prikaz>
<prikaz> $\Longrightarrow$ <popis> <prikaz>
<prikaz> $\Longrightarrow$ `\caption` <nepovinne> <popis> <prikaz>
<prikaz> $\Longrightarrow$ ε
<popis> $\Longrightarrow$ {<body>}
<nepovinne> $\Longrightarrow$ [<body>]
<nepovinne> $\Longrightarrow$ ε
<sloupce> $\Longrightarrow$ {<def-sloupce>}
<def-sloupce> $\Longrightarrow$ kombinace cisel, znaku c, l, r, @{, prikazu, }, * a || a > {`\columncolor`
<barva2> <procento> } a <del-reg>
<tabular> $\Longrightarrow$ <blok-textu-tabular> <dalsi-sloupce> <tabular>
<tabular> $\Longrightarrow$ ε
<dalsi-sloupce> $\Longrightarrow$ & <blok-textu-tabular> <dalsi-sloupce>
<dalsi-sloupce> $\Longrightarrow$ ε
<blok-textu-tabular> $\Longrightarrow$ <tabular-prikaz> <body> <tabular-prikaz>
<tabular-prikaz> $\Longrightarrow$ ε
<tabular-prikaz> $\Longrightarrow$ `\hline`
<tabular-prikaz> $\Longrightarrow$ `\cline`{cislo-cislo}
<tabular-prikaz> $\Longrightarrow$ `\scalebox` <procento>
<tabular-prikaz> $\Longrightarrow$ `\resizebox` <prumer><prumer>
<tabular-prikaz> $\Longrightarrow$ `\rotatebox` <procento>
<tabular-prikaz> $\Longrightarrow$ `\reflectbox` <popis>
<tabular-prikaz> $\Longrightarrow$ `\rowcolor` <barva2><procento>
<tabular-prikaz> $\Longrightarrow$ <table>

Dodatek B

Popis příkazů

Příkazy jsou tematicky rozděleny podle oblastí, kde se mohou nejčastěji vyskytovat. Tato část je víceméně převzata z knihy Jiřího Rybičky ([3]).

Barvy

\color{barva} - příkaz přepne barvu následujícího textu

\colorbox{barva}{text} - příkaz vypíše text v barevném obdélníku

\columncolor[model]{barva} - barva tabulkového sloupce v barevném modelu, příkaz je součástí balíku colortbl

\definecolor{barva}{model}{h} - definuje novou barvu v určitém barevném modelu (rgb anebo gray). Parametr h definuje hodnoty barevných složek nebo úrovně šedi

\fcolorbox{barR}{barI}{text} - vytvoří rámeček barvou barR, vnitřek vyplní barvou barI a textem

\pagecolor{barva} - nastaví se barva pozadí na celé stránce

\rowcolor[model]{barva} - nastavení barvy v řádku tabulky v barevném modelu, je součástí balíku colortbl

\textcolor{barva}{text} - nastaví barvu pro text v závorkách

Nerámované boxy

\depth - délkový registr nabývající hloubky boxu. Je použitelný pouze ve volitelném parametru příkazů `\makebox`, `\framebox` a `\raisebox`.

\height - délkový registr udávající výšku zpracovaného boxu, omezení stejné jako u `\depth`

\makebox[šířka][pozice]{text} - příkaz pro vytvoření boxu zadané šířky, v němž je text umístěn podle specifikace pozice vpravo (r), vlevo (l) nebo doprostřed (pozice zůstane prázdná). Není-li udána šířka, je box široký na délku textu.

lrbox{box} - prostředí pro úschovu svého obsahu do box.

\mbox{text} - příkaz pro vytvoření boxu o šířce dané textem, zjednodušená varianta `\makebox`.

minipage - prostředí pro vytvoření boxu, který se chová jako stránka.

newsavebox{příkaz} - definuje příkaz jako úschovnou oblast pro úschovu boxů.

parbox[**p**][**výška**][**vn**]{**šířka**}{**text**} - vytvoří box o zadané šířce, do něhož se umístí text v odstavcovém režimu. Nepovinný parametr p udává pozici parboxu vzhledem k okolí] t] zarovnání na horní okraj, b] zarovnání na dolní okraj. Implicitně je zarovnáván na střed. Výška definuje požadovanou výšku, parametr vn způsob dosazení obsahu do boxu.

raisebox{z}[**výška**][**hloubka**]{**text**} - vytvoří box s textem, který bude zvýšen o z nad základnu. Parametr výška a hloubka definují kolik místa má box zabírat nahoře a dole.

rule[**zvýš** {šířka}{výška}] - vykreslení černého boxu o rozměrech šířka a výška, který bude umístěn o zvýš nad základnu.

savebox{příkaz}[**šířka**]{**box**} - umístí do úschovné oblasti příkazem definovaný box, který má mít šířku. Příkaz musí již být definovaný příkazem `\newsavebox`. Není-li šířka uvedena, je použita šířka daného boxu.

sbox{příkaz}{box} - stejná funkce jako `\savebox` v textovém režimu, ale bez volitelných parametrů.

totalheight - délkový registr - celková výška právě zpracovaného boxu

usebox{příkaz} - příkaz pro vysázení boxu uschovaného dříve do příkazu

width - délkový registr udávající šířku zpracovaného boxu

Rámované boxy

fbox{text} - příkaz pro vložení textu do rámečku.

fboxrule - míra definující tloušťku čáry pro rámečky v příkazu `\fbox`.

fboxsep - míra definující vzdálenost textu od rámečku v příkazu `\fbox`.

framebox[**šířka**][**pozice**]{**text**} - příkaz pro orámování textu. Vytvoří se box o dané šířce, text je zobrazen podle pozice] l] vlevo, r] vpravo, s] rozptýlit. Při vynechání pozice se text vystředí

Čítače

addtocounter{čítač}{hodnota} - přičtení hodnoty do čítače

alph{čítač} - výpis hodnoty čítače odpovídající malému písmenu (1 *rightarrow* a)

Alph{čítač} - výpis hodnoty čítače odpovídající velkému písmenu (1 *rightarrow* A)

arabic{čítač} - výpis hodnoty čítače v podobě arabských číslic

newcounter{čítač}[**nadřízený**] - definuje nový čítač, nepovinný parametr nadřízený udává jméno již exitujícího čítače, jehož zvýšením o 1 pomocí příkazů `\stepcounter` anebo `\refstepcounter` se nuluje hodnota čítače.

refstepcounter{čítač} - zvýší hodnotu čítače o 1 a nuluje všechny závislé čítače. Definuje hodnotu, kterou lze získat příkazem `\ref`.

roman{čítač} - výpis hodnoty čítače malými římskými číslicemi.

Roman{čítač} - výpis hodnoty čítače velkými římskými číslicemi.

setcounter{čítač}{hodnota} - nastavení čítače na hodnotu

stepcounter{čítač} - zvýší hodnotu čítače o 1 a nuluje všechny závislé čítače.

thečítač - příkaz vzniká automaticky se vznikem daného čítače, vydává hodnotu čítače ve formě textu.

usecounter{čítač} - příkaz pro definici čítače, jehož pomocí budou v nově definovaném výčtovém prostředí číslovány položky.

value{čítač} - příkaz pro získání hodnoty čítače.

Dělení slov

- - uvnitř slova označuje místo, kde má dojít k dělení, v jiných místech slovo děleno nebude.

czech - přepnutí jazyka na češtinu.

english - přepnutí jazyka na angličtinu.

hyphenation{seznam slov} - příkaz pro výčet výjimek v dělení slov. Seznam slov obsahuje slova oddělená mezerami, slova obsahují pomlčky naznačující dělení.

lefthyphenmin - registr pro nastavení minimálního počtu písmen v první části děleného slova.

nosplit - volba balíku czech a slovak, globálně potlačí opakování spojovníku na následujícím řádku.

righthyphenmin - minimální počet znaků ve druhé části děleného slova.

slovak - přepnutí jazyka na slovenštinu.

split - volba balíku czech a slovak, nařídí globálně opakování spojovníku na následujícím řádku.

splithyphens - způsobí opakování spojovníku na následujícím řádku.

standarthyphens - zamezí opakování spojovníku na následujícím řádku.

Délkové registry

addtolength{délka}{velikost} - přidá hodnotu velikost k délkovému registru délka.

minus - specifikace stažení pružné délky.

newlength{reg} - příkaz pro vytvoření délkového registru reg.

plus - specifikace roztažení pružné délky.

setlength{reg}{velikost} - nastavení délkového registru reg na velikost.

settodepth{reg}{text} - nastavení hloubkového registru reg na velikost hloubky, kterou zabírá text.

settoheight{reg}{text} - nastavení délkového registru reg na velikost výšky, kterou zabírá text.

settowidth{reg}{text} - nastavení délkového registru reg na velikost, kterou zabírá text.

Dokument

document prostředí dokumentu.

documentclass[v]**{třída}{dt}** - uvozuje zdrojový text, definuje použitou třídu sazby, její volby v. Je-li datum vytvoření dané verze třídy dřívější než dt, je vypsáno varování.

filecontents{soubor} - prostředí zapisuje pomocné identifikační údaje do souboru] musí být uveden před příkazem `\documentclass`.

documentstyle[**volby**]**{styl}** - příkaz uvozuje zdrojový text. Definuje styl sazby, seznam jmen volby představuje modifikace stylu, případně jména dalších souborů s příkazy, které se čtou před zpracováním textu v dokumentu.

listfiles - příkaz (pouze v hlavičce), vypíše seznam souborů nutných pro zpracování daného dokumentu.

usepackage[**v**]**{b}**[**dt**] - připojení balíku b, s datem verze mladším než dt a případnými volbami v.

Dopisy - letter

address{adresa} - příkaz pro definici zpáteční adresy.

cc{osoby} - definuje seznam osob, kterým jsou zaslány kopie dopisu.

closing{pozdrav} - závěrečný pozdrav.

encl{seznam příloh} - vysází seznam příloh.

makelabels - v preambuli třídy letter způsobí tisk adres všech prostředí letter v souboru.

opening{pozdrav} - vysázení pozdravu v dopise.

ps - uvádí část Post Scriptum.

signature{jméno} - definuje jméno vypisované na konci dopisu.

Doplňky tříd a balíků

11pt - volba pro základní velikost písma 11pt.

12pt - volba pro základní velikost písma 12pt.

a4paper - volba pro sazbu stránek velikosti A4 (210×297 mm).

a5paper - volba pro sazbu stránek velikosti A5 (148×210 mm).

b5paper - volba pro sazbu stránek velikosti B5 (176×250 mm).

executivepaper - volba pro sazbu stránek velikosti 7,25×10,5 palce.

fleqn - sazba vysazených vzorců k levému kraji.

landscape - vzájemně zamění definovanou šířku a výšku stránky.

legalpaper - volba pro sazbu stránek velikosti 8,5×14 palců.

letterpaper - volba pro sazbu stránek velikosti 8,5×11 palců.

mathindent - odsazení vzorců od levého okraje při stylové volbě fleqn dosazující vzorce na levý okraj

notitlepage - znemožnění práce příkazu `\maketitle` pro sazbu titulní strany, implicitně nastavena ve třídě article

oneside - sazba všech stránek pravostranně.

twoside - rozlišování pravých a levých stránek; oboustranný dokument.

openany - zahájení kapitoly na libovolné stránce (pravé i levé).

openright - zahájení kapitoly pouze na prevé (liché) stránce.

multicol - balík pro sazbu do dvou sloupců.

twocolumn - volba pro sazbu do dvou sloupců.

Doslovný text

alltt - prostředí, text je chápán jako v prostředí verbatim, ale znaky \, { a } mají svůj obvyklý význam.

verbznak text znak - příkaz pro doslovné vysázení textu ohraničeného zvoleným znakem.

verb*znak text znak - totéž co příkaz `\verb` ale mezery se sází s pevnou délkou čtverčíku.

verbatim - prostředí pro doslovnou sazbu textu

verbatim* - totéž co prostředí verbatim, ale mezery se sází s pevnou délkou čtverčíku

Dva a více sloupců

columnsep - míra udávající mezeru mezi sloupci v dvousloupcovém režimu.

columnseprule - míra udávající tloušťku čáry mezi sloupci.

multicol - balík pro vícesloupcovou sazbu s rozšířenými možnostmi než ve stylu twocolumn

multicols{sl}[nad]{sk} - prostředí pro ohraničení úseku vícesloupcové sazby s počtem sloupců sl. Před začátkem vícesloupcové sazby může být nadpis nad, na něm nenastane stránkový zlom. Začátek prostředí včetně nadpisu se umístí na novou stránku, zbývá-li do konce běžné stránky méně než sk. Parametry se uvádějí pouze u začátečního příkazu prostředí. Prostředí je definováno v balíku multicol.

onecolumn - příkaz pro přechod z dvousloupcového do jednosloupcového režimu (při volbě stylu twocolumn).

twocolumn{text} - příkaz pro start dvousloupcového režimu. Je-li uveden volitelný parametr, umístí se text v šířce přes dva sloupce před začátek dvousloupcového režimu.

Fonty

bfdefault - nastavení pro `\bfseries` a `\textbf`

DeclareFontFamily{k}{r}{opt} - definice nové rodiny r písma daného kódu k.

DeclareFontShape{k}{r}{v}{t}{s}{opt} - definice člena deklarované rodiny r.

encodingdefault - kódování základního písma.

familydefault - rodina základního písma.

font \ **jméno=soubor zvětšení** - primitiv TeXu pro zavedení nového fontu ze souboru. Nové jméno lze pak použít podobně jako např. `\rm`. Zvětšení lze specifikovat klauzulí `at` míra (přímé zadání kuželky), klauzulí `scaled` násobek (celé číslo udávající násobek základní velikosti, základní velikost = 1000) anebo klauzulí `\magstep` číslo (rozmezí 0-5) nebo `\magstephalf`.

fontencoding{kódování} - nastavení kódování fontu.

fontfamily{rodina} - nastavení rodiny písma.

fontseries{váha} - nastavení duktu písma.

fontshape{tvar} - nastavení tvaru písma.

fontsize{stupeň}{řádkování} - nastavení stupně písma.

itdefault - nastavení pro `\itshape` a `\textit`.

load{velikost}{styl} - zavedení fontu určitého stylu a velikosti v matematickém režimu.

magstep - stupeň zvětšení fontu. Následující číslo - 0 - základní velikost, 1 - zvětšení 1,2× atd

magstephalf - příkaz zvětšení o polovinu prvního stupně (písmo 11pt).

mddefault - nastavení pro `\mdseries` a `\textmd`.

newfont{příkaz}{jméno} - definice nového fontu. Font bude dostupný pomocí příkazu a jedná se o fontový soubor se jménem jméno.

rmdefault - nastavení pro `\rmfamily` a `\textrm`.

scdefault - nastavení pro `\scfamily` a `\textsc`.

seriesdefault - váha základního písma.

sfdefault - nastavení pro `\sffamily` a `\textsf`.

shapedefault - tvar základního písma.

sldefault - nastavení pro `\slshape` a `\textsl`.

ttdefault - nastavení pro `\ttfamily` a `\texttt`.

updefault - nastavení pro `\upshape` a `\textup`.

Glosář

glossary{heslo} - příkaz pro zavedení hesla do souboru .glo pro tvorbu glosáře.

glossaryentry - příkaz, který se objeví v souboru .glo jako efekt příkazu `\glossary`.

makeglossary - příkaz způsobující vytvoření souboru .glo činností příkazů `\glossary`.

Grafika

graphpaper[**m**](**poč**)(**v**) - příkaz balíku graphpap vykreslí mřížku počínaje souřadnicemi poč s velikostí v. Rozteč mřížky je každých 10 jednotek, nebo podle zadání parametru m. Např. `\graphpaper[5](5, 8)(40, 50)`.

includegraphics[**ld**][**ph**]**{j}** - vloží do textu obrázek uložený do souboru j, rozměry boxu budou podle volitelných parametrů.

includegraphics*[**ld**][**ph**]**{j}** - vloží do textu obrázek uložený do souboru j, z něhož vytvoří výřez podle volitelných parametrů.

reflectbox{box} - zrcadlově převrátí box v horizontálním směru.

resizebox{šířka}{výška}{box} - roztáhne (stlačí) box na rozměry šířka a výška. Je-li nějaký parametr nahrazen znakem ! je odpovídající rozměr doplněn proporcionálně podle rozměru druhého.

resizebox*{**šířka**}{**výška**}{**box**} - stejně jako předešlé, ale výška zahrnuje součet výšek nad a pod účařím.

rotatebox{úhel}{obsah} - otočení obsahu o zadaný úhel ve stupních proti směru hodinových ručiček.

scalebox{faktor}[**faktory**]**{box}** - zvětší rozměry boxu faktor. Volitelný parametr faktory umožňuje zvlášť upravit rozměr ve svislém směru.

Indexy a exponenty

Indexy a exponenty se smí vyskytnout pouze v matematickém prostředí.

_ - znak pro sazbu indexu.

^ - znak pro sazbu exponentu.

Kreslení obrázků - picture

\bezier{N}(A)(B)(C) - nakreslení Bezierovy křivky z bodu A do bodu C s řídícím bodě v B, která je složena z N bodů.

\circle{průměr} - vykreslí kružnici daného průměru.

\circle*{průměr} - vykreslí plný kruh daného průměru.

\dashbox{elem}(š, v)[pozice]{text} - čárkovaný obdélník, čára je složena z elementů délky elem, a jehož rozměry jsou š a v. Text se umístí podle pozice.

\frame{objekt} - orámování objektu, v prostředí picture.

\framebox(výška, šířka)[pozice]{obsah} - rámeček zadaných rozměrů, v němž je obsah umístěn podle pozice.

\line(x, y){délka} - příkaz pro kreslení úsečky

\linethickness{tloušťka} - nastavení tloušťky vodorovných a svislých čar.

\makebox(šířka, výška)[pozice]{objekt} - v prostředí picture vytvoří box dle zadané specifikace.

\multiput(x, y)(dx, dy){p}{o} - opakování objektu o počínaje v souřadnicích x, y a rozdílem pozic dx, dy. Opakuje se p-krát.

\oval(šířka, výška)[část] - vykreslení oválu o zadaných parametrech. Pokud je uveden nepovinný údaj část, vykreslí se jen požadovaná část oválu (nejméně ale čtvrtina).

\put(xr, yr){objekt} - umístí na pozice xr a yr obrazový objekt.

\qbezier[N](A)(B)(C) - není-li N uvedeno, je beziérova křivka kreslena plnou čarou.

\qbeziermax - příkaz definuje maximální počet bodů pro kreslení Beziérovy křivky; lze měnit pomocí `\renewcommand`.

\savebox{př}(x, y)[m]{o} - úschova zobrazovaného objektu o do boxu x, y. Způsob umístění m je totožný s možnostmi u `\makebox`.

\shortstack{řádky} - výpis argumentu řádky nad sebou (řádky jsou odděleny pomocí \\); není v prostředí picture

\thicklines - přepnutí do režimu tlustších čar.

\thinlines - přepnutí do režimu tenších čar.

\unitlength - délkový registr představující jednotku pro všechny míry v prostředí picture.

\vector(dx, dy){délka} - vysázení úsečky se šipkou na konci. Sklon je dán poměrem dx a dy a délka je dána parametrem délka.

Loga

Vykreslí příslušné loga.

LaTeX - LaTeX

LaTeXe - LaTeX 2$_\varepsilon$

TeX - TeX

Matematická prostředí

$ - znak začínající a končící matematické prostředí uvnitř odstavce.

$$ - začátek a konec matematického prostředí uvnitř odstavce.

\[- začátek vysazeného matematického prostředí.

\] - konec vysazeného matematického prostředí.

displaymath - prostředí pro vysazený matematický text.

\(- začátek matematického prostředí uvnitř odstavce.

\) - konec matematického prostředí uvnitř odstavce.

math - matematické prostředí pro vzorce uvedené uvnitř odstavce.

eqnarray - matematické prostředí pro sazbu posloupnosti rovnic s číslováním řádků.

equation - matematické prostředí pro sazbu číslovaného vztahu, anebo čítač spojený s tímto prostředím, který obsahuje aktuální číslo rovnice.

Horizontální mezery v textovém režimu

\␣ - mezislovní mezera

, - zúžená mezera, 1/6 čtverčíku

\/ - mezera pro kurzivní korekci.

@ - uvedeno před tečkou, značí, že tečka ukončuje větu] je totiž za ní jiná mezera než mezislovní.

frenchspacing - řízení mezerování po tečce na konci věty.

nofrenchspacing - příkaz upravující způsob mezerování po konci věty.

~ - nezlomitelná mezera.

hfill - příkaz pro vodorovnou pružnou mezeru maximální možné velikosti.

hspace{míra} - příkaz pro vodorovnou mezeru, není proveden na začátku nebo na konci řádku.

hspace*{míra} - je uveden i na začátku nebo na konci řádku.

stretch{číslo} - pružná délka s přirozenou velikostí 0 a roztažitelností číslo. Argument je reálné číslo s volitelným znaménkem představující násobek roztažitelnosti délky `\fill`.

Horizontální mezery v matematickém režimu

! - záporná úzká mezera

; - široká mezera

\␣ - mezislovní mezera

, - úzká mezera

: - střední mezera

quad - čtverčík

qquad - dva čtverčíky

Vertikální mezery v textu

addvspace{míra} - zvětšení vertikální mezery o míra.

\\[**míra**] - nepodmíněný konec řádku, volitelný parametr způsobí vertikální mezeru k následujícímu řádku.

***[míra**] - nepodmíněný konec řádku, po němž nemá nastat zlom stránky. Nepovinný parametr způsobí vertikální mezeru k následujícímu řádku.

bigskip - vertikální mezera o velikosti `\bigskipamount`.

bigskipamount - mezera, jejíž velikost je dána použitým stylem (přibližně výška řádku).

medskip - příkaz pro vertikální mezeru velikosti `\medskipamount`.

medskipamount - přibližně polovina výšky řádku.

smallskip - vertikální mezera velikosti `\smallskipamount`.

smallskipamount - přibližně čtvrtina výšky řádku.

vfill - svislá roztažitelná mezera, ekvivalentní k `\vspace{\fill}`.

vspace{míra} - vertikální mezera velikosti míra, mezera se nevytvoří na začátku nebo na konci stránky.

vspace*{míra} - mezera se vytvoří i na začátku i na konci stránky.

Vertikální mezery před vzorci

abovedisplayshortskip - vertikální mezera mezi krátkým vzorcem a předchozím textem. Krátký vzorec je takový, který začíná vpravo od místa, kde končí předchozí řádek; opakem je dlouhý vzorec.

abovedisplayskip - vertikální mezera mezi dlouhým vzorcem a předchozím textem.

belowdisplayskip - vertikální mezera pod dlouhým vzorcem.

belowdisplayshortskip - vertikální mezera pod krátkým vzorcem.

Měrné jednotky

cc - délková jednotka cicero.

cm - délková jednotka centimetr.

dd - délková jednotka Didôtův bod.

em - délková jednotka závislá na zvoleném písmu, je rovna stupni písma (přibližně odpovídá "M").

ex - délková jednotka závislá na zvoleném písmu, přibližně odpovídá "x".

in - délková jednotka palec.

mm - délková jednotka milimetr.

pc - délková jednotka pica.

pt - délková jednotka point.

sp - délková jednotka scaled point.

Nadtržení

overline{text} - sazba vodorovné čáry nad textem (pouze v matematickém režimu).

Nová stránka

cleardoublepage - přechod na další lichou stránku (ve dvoustranném režimu), vysazení všech dosud nevysázených plovoucích objektů.

clearpage - přechod na další stránku, vysazení všech dosud nevysázených plovoucích objektů.

newpage - nová stránka, vysazení všech dosud nevysázených plovoucích objektů.

nopagebreak[**stupeň**] - zakazuje stránkový zlom v daném místě, stupeň je celé číslo 0-4, udává sílu příkazu (4] nejsilnější, implicitně).

pagebreak[**stupeň**] - stránkový zlom v daném místě, stupeň podobně jako výše.

samepage - zamezuje stránkový zlom v rozsahu své působnosti.

Nový řádek

\\[**míra**] - nepodmíněný konec řádku, volitelný parametr způsobí mezeru k následujícímu řádku.

***[**míra**] - nepodmíněný konec řádku, po němž nemá nastat zlom stránky. Nepovinný parametr způsobí vertikální mezeru k následujícímu řádku.

linebreak[**stupeň**] - řádkový zlom, stupeň udává sílu příkazu 0-4, bez parametru je implicitně 4.

newline - začátek nového řádku, současný řádek nebude zarovnám na pravém okraji.

nolinebreak[**stupeň**] - zabrání řádkovému zlomu, parametr stejný jako u `\linebreak`.

Obsah

addcontentsline{ext}{ú}{t} - přidá do obsahového souboru s rozšířením ext text t, jehož logická úroveň má být ú. Např.

```
\addtocontentsline\{toc\} \{section\} \{Úvod\}
```

addtocontents{ext}{text} - přidá text do obsahového souboru s rozšířením ext.

listoffigures - vytvoření seznamu obrázků.

listoftables - vytvoření seznamu tabulek.

numberline{číslo sekce}{titulek} - vypsání řádku do obsahu; vypíše se text do čísla sekce a titulek a doplní se správné číslo stránky.

tableofcontents - vysázení obsahu.

tocdepth - čítač udávající, která úroveň titulků bude zanášena automaticky do obsahu (1, sekce, 2, subsekce, atd.)

Oddíly textu

appendix - příkaz začínající dodatky. Způsobí změnu číslování titulků nejvyšší úrovně.

backmatter - příkaz ve třídě book uvozující zadní část knihy (literatura, rejstřík, tiráž). Příkaz `\chapter` je zde automaticky v nečíslované podobě i bez uvedení hvězdičky.

chapter - čítač obsahující aktuální číslo kapitoly.

chapter[obsah]{nadpis} - příkaz pro titulek kapitoly, parametr obsah definuje text obsahové položky, nadpis je nadpis kapitoly v textu. Nadpis je v textu označen číslem kapitoly, které je použito z čítače chapter. Pak je tento čítač automaticky zvýšen o jedničku.

chapter*{nadpis} - nečíslovaná podoba nadpisu kapitoly.

frontmatter - uvozuje přední část knihy (ve třídě book). Příkaz `\chapter` jeho vlivem nebude číslovat následující části.

mainmatter - příkaz uvozující hlavní textovou část knihy.

paragraph - čítač obsahující aktuální číslo odstavce.

paragraph[obsah]{titulek} - příkaz pro titulek odstavce. Pokud je uveden nepovinný parametr, je do obsahu zanesen text obsah, jinak text titulek. Číslo odstavce je použito z čítače paragraph, tento čítač je poté automaticky zvýšen o jedničku.

paragraph*{titulek} - příkaz pro nečíslovaný titulek odstavce.

part - čítač obsahující aktuální číslo části textu (nejvyšší oddíl textu - partu).

part - vytvoření titulku o důležitosti vyšší než kapitola (sekce), parametry i činnosti příkazu.

secnumdepth - čítač ovlivňující číslování titulků. Čím větší hodnota, tím více úrovní titulků se čísluje. 1 - číslování sekcí, 2 - číslování sekcí a podsekcí, atd.

section - čítač obsahující aktuální číslo sekce textu.

section - příkaz pro titulek důležitosti sekce; `\section*`

subparagraph - čítač spojený s titulkem úrovně pododstavce.

subparagraph - příkaz pro titulek úrovně pododstavce, parametry a chování jsou obdobou příkazu `\chapter`, včetně formy s hvězdičkou.

subsection - čítač spojený s titulkem úrovně podsekce.

subsection - příkaz pro titulek úrovně podsekce, parametry a chování jsou obdobou příkazu `\chapter`, včetně formy s hvězdičkou.

subsubsection - čítač spojený s titulkem úrovně podpodsekce.

subsubsection - příkaz pro titulek úrovně podpodsekce, parametry a chování jsou obdobou příkazu `\chapter`, včetně formy s hvězdičkou.

Odkazy v textu

label{návěští} - definuje hodnotu návěští, na niž se odvolávají příkazy `\ref`, případně `\pageref`.

pageref{návěští} - produkuje číslo stránky, na které se nachází příkaz `\label{návěští}`.

ref{návěští} - příkaz produkuje text spojený s návěštím. Návěští je definováno příkazem `\label` a podle místa definice se s návěštím spojí číslo běžného oddílu, číslo tabulky, obrázku a podobně.

Ochrana parametrů

protect - chrání "křehký" příkaz v pohyblivém argumentu jiného příkazu.

Stupeň

Příkazy pro změnu velikosti písma.

tiny - ^{ukázkový text}

footnotesize - ukázkový text

scriptsize - ukázkový text

small - ukázkový text

normalsize - ukázkový text

large - ukázkový text

Large - ukázkový text

LARGE - ukázkový text

huge - ukázkový text

Huge - ukázkový text

Stupeň - matematika

displaystyle - změna stupně písma na vysazenou (největší) velikost.

scriptstyle - velikost prvních indexů.

scriptscriptstyle - velikost druhých indexů.

textstyle - velikost odpovídající formuli umístěné uvnitř odstavce.

Typ, řez

bf - tučné písmo.

bfseries - nastavení atributu váhy písma.

em - zdůrazněné písmo.

emph{text} - příkaz pro zdůraznění textu.

fontencoding{kódování} - nastavení kódování fontu.

fontfamily{rodina} - nastavení rodiny písma.

fontseries{váha} - nastavení duktu písma.

fontshape{tvar} - nastavení tvaru písma.

fontsize{stupeň}{řádkování} - nastavení stupně písma.

it - kurzíva

itshape - nastavení atributu písma sklon.

mdseries - nastavení střední váhy písma.

normalfont - nastavení atributů písma váha, rodina, sklon na implicitní hodnoty příslušející hlavnímu písmu.

rm - antikva.

rmfamily - nastavení rodiny písma na antikvu.

sc - kapitálky.

scshape - příkaz pro nastavení atributu tvar na small caps.

selectfont - volba nastaveného písma.

sf - bezserifové písmo.

sffamily - příkaz pro nastavení rodiny písma na sans serif.

sl - skloněné písmo.

slshape - příkaz pro nastavení sklonu písma.

textbf{text} - nastaví u textu atribut váha na tučně.

textit{text} - nastaví u textu atribut sklon na kurzívu.

textmd{text} - nastaví textu atribut tučnost na střední (implicitní).

textnormal{text} - nastaví u textu atributy na normální písmo.

textrm{text} - nastaví u textu atribut rodina na antikvu.

textsc{text} - nastaví u textu atribut rodina na small caps.

textsf{text} - nastaví u textu atribut rodina na sans serif.

textsl{text} - nastaví u textu atribut sklon na slanted.

texttt{text} - nastaví u textu atribut rodina na strojopisné písmo.

textup{text} - nastaví u textu atribut sklon na vzpřímený.

tt - strojopisné písmo.

ttfamily - nastavení atributu rodina na strojopisné písmo.

upshape - nastavení atributu sklon na vzpřímené písmo.

Typ, řez - matematika

boldmath - deklarace nařizující sazbu tučných symbolů a písmen v matematickém režimu (musí být uvedena v textu před matematickým prostředím).

cal - kaligrafické písmena v matematickém režimu AB

DeclareMathSymbol{p}{k}{f}{č} - přiřazení kategorie k matematickému symbolu znaku daného čísla č z fontu f symbolů. Symbol bude použitelný zápisem příkazu p.

DeclareSymbolFont{f}{k}{r}{v}{t} - zavedení fontu f dané rodiny r, kódování k, váhy h a tvaru t jako fontu matematických symbolů.

mathalpha - kategorie matematického symbolu - abecední znak.

mathbf{text} - tučné písmo v matematickém režimu.

marhbin - kategorie matetického symbolu] binární operátor.

mathcal{text} - kaligrafické písmo v matematickém režimu.

mathclose - kategorie matematického symbolu] uzavření.

mathit{text} - kurzíva v matematickém režimu.

mathop - velký operátor.

mathopen symbol - otevření.

mathord symbol - obyčejný symbol.

mathpunct - interpunkce.

mathrel symbol - relační operátor.

mathrm{text} - antikva v matematickém režimu.

mathsf{text} - sans serif v matematickém režimu.

mit - příkaz pro změnu písma na matematickou italiku.

unboldmath - příkaz přepínající v matematickém režimu na sazbu netučných symbolů a proměnných.

Plovoucí objekty

bottomfraction - maximální dolní část stránky, která může být obsazena plovoucími objekty.

bottomnumber - čítač udávající maximální počet plovoucích objektů v dolní části stránky.

caption[**obsah**]{**popisek**} - příkaz pro popisek obrázku nebo tabulky. Použití v prostředí figure nebo picture. Nepovinný parametr obsah definuje text zanesený do seznamu tabulek.

dblfloatpagefraction - minimální část stránky obsazené plovoucími objekty v dvousloupcovém režimu.

dblfloatsep - vzdálenost mezi plovoucími objekty ve dvou sloupcovém režimu.

dbltextfloatsep - vzdálenost mezi plovoucím objektem a textem ve dvousloupcovém režimu.

dbltopfraction - maximální část stránky, která může být věnována plovoucím objektům v dvousloupcovém režimu.

dbltopnumber - čítač udávající maximální počet plovoucích objektů v horní části stránky ve dvousloupcovém režimu.

figure[**umístění**] - plovoucí prostředí pro obrázky.

figure - čítač spojený s prostředím figure, je obsluhován pomocí příkazu caption a obsahuje aktuální číslo obrázku.

figure*[**umístění**] - plovoucí prostředí pro obrázky přes celou šířku stránky v dvousloupcovém režimu.

floatpagefraction míra udávající, jakou část stránky musí minimálně zabírat plovoucí objekty na samostatné stránce.

floatsep - míra udávající vzdálenost mezi plovoucími objekty na samostatné stránce.

intextsep - vertikální mezera vložená nad a pod plovoucí objekt, který je umístěn doprostřed textu.

normalmarginpar - vysázení okrajových poznámek na implicitním okraji (vpravo, nebo vně).

suppressfloats[**umístění**] - potlačí umístění plovoucích prostředí s daným umístěním (t nebo b) na aktuální stránce

table - čítač v prostředím table pro číslování tabulek. Obsluhován příkazem `\caption`.

table[**umístění**] - plovoucí prostředí pro tabulky.

table*[**umístění**] - umístění tabulky přes šířku stránky přes všechny sloupce ve vícesloupcovém prostředí.

textfloatsep - míra udávající svislou vzdálenost plovoucího objektu od textu.

textfraction - minimální podíl textu na stránce s umístěným plovoucím objektem.

topfraction - číslo udávající část stránky, která může být maximálně věnována plovoucímu objektu umístěnému v horní části stránky.

topnumber - čítač udávající maximální počet plovoucích objektů, které lze umístit do horní části stránky.

totalnumber - čítač udávající maximální počet plovoucích objektů, které se mohou objevit na jedné stránce.

Podtržení

underline{text} - příkaz pro podtržení textu (v matematickém i textovém režimu).

Pomlčky

texttemdash - čtverčíková pomlčka.

texttendash - půlčtverčíková pomlčka.

Poznámka ve zdrojovém textu

% - text za tímto znakem až do konce řádku je chápán jako komentář.

Poznámka na okraji

marginpar[**levá**]**{poznámka}** - příkaz pro poznámku na okraji, nepovinný parametr se vysází jen tehdy, je-li příkaz nařízen ve dvoustranném režimu a po poznámka se nachází na levém okraji. V jiných případech se vysází poznámka.

marginparpush - vertikální mezera mezi dvěma okrajovými mezerami.

marginparsep - mezera mezi okrajem textu a okrajovou poznámkou.

marginparwidth - šířka okrajové poznámky.

reversemarginpar - příkaz, okrajové poznámky budou vysázeny na opačném kraji (vlevo, případně na vnitřním okraji).

Poznámka pod čarou

footnote čítač obsahující aktuální číslo poznámky pod čarou.

footnote[**číslo**]**{poznámka}** - generuje poznámku pod čarou s textem poznámka. Je-li uveden parametr číslo, je použit pro očíslování poznámky, v opačném případě se použije hodnota čítače footnote, který se předtím zvýší o jedničku.

footnotemark[**číslo**] - generuje odkazové číslo na poznámku.

footnoterule - generuje čáru pro poznámky pod čarou.

footnotesep - velikost svislého odskoku, který se použije před vysázením textu poznámky. Určuje vzdálenost poznámek od sebe navzájem.

footnotetext[**číslo**]**{poznámka}** - ve spojení s příkazem `\footnotemark` vysází text poznámky, která má číslo. Čítač footnote není zvýšen.

mpfootnote - čítač, který obsahuje aktuální číslo poznámky pod čarou v prostředí minipage.

Přímé pokyny pro zobrazovač

special - příkaz, jehož argument nebude překládán, ale bude beze změny přenesen do souboru .dvi.

Rejstřík

index{heslo} - zařazení hesla do souboru .idx pro tvorbu rejstříku.

indexentry - příkaz, který se objeví v souboru .idx jako důsledek činnosti příkazu `\index`.

indexspace - vytvoří v rejstříku svislou mezeru.

makeidx - balík pro sazbu rejstříku.

makeindex - vytvoření souboru .idx činností příkazů `\index`.

printindex - příkaz pro umístění rejstříku do textu (definován v balíku makeidx).

subitem - příkaz představující položku rejstříku druhé úrovně v prostředí theindex.

subsubitem - příkaz představující položku rejstříku třetí úrovněv prostředí theindex.

theindex - prostředí pro sazbu rejstříku.

Řádkování

baselineskip - vzdálenost účaří dvou po sobě jdoucích řádků v jednom odstavci.

baselinestretch - číslo, kterým se násobí míra `\baselineskip`. Jeho změnou lze dosáhnout různé hustoty řádkování.

Sazba odstavců

fussy - příkaz navracející způsob sazby odstavce do běžného stavu (po příkazu `\sloppy`).

indent - příkaz pro zahájení odstavce s odstavcovou zarážkou.

leftskip - délkový registr pro levý okraj odstavce.

linewidth - míra udávající aktuální šířku textu. Její velikost nesmí být měněna.

noindent - příkaz pro zahájení odstavce bez odstavcové zarážky.

par - příkaz pro začátek nového odstavce, je ekvivalentní prázdnému řádku.

parindent - míra udávající velikost odstavcové zarážky.

parskip - míra udávající vertikální vzdálenost odstavců v běžném textu.

raggedleft - příkaz, sazba odstavců se zarovnáním na pravý okraj (levý nezarovnán).

raggedright - příkaz sazba odstavců se zarovnáním na levý okraj (pravý nezarovnán).

rightskip - délkový registr pro pravý okraj odstavce.

sloppy - příkaz pro nařízení méně přesné sazby odstavců (větší mezery mezi slovy), aby se zabránilo přetékání pravého okraje při zarovnávání. Viz též `\fussy`.

sloppypar - prostředí, v němž jsou odstavce sázeny méně přesně] příkazem `\sloppy`.

Sazba stránky

enlargethispage{délka} - zvětší výšku dané stránky o délku.

enlargethispage*{délka} - zároveň zmenší vertikální mezery dané stránky na nejnižší možnou velikost.

flushbottom - deklarace zajišťující vyplnění každé stránky na stejnou výškovou velikost. Do míst mezi odstavci se automaticky vkládají příslušné mezery.

hoffset - délkový registr umožňující horizontální posunutí sazebního obrazce na stránce.

markboth{vlevo}{vpravo} - příkaz pro dosazení textů do běžných hlaviček ve dvoustranném textu.

markright{text} - dosazení textu do běžných hlaviček jednostranného textu.

page - čítač s hodnotou čísla stránky.

pageheight - míra udávající výšku listu.

pagenumbering{způsob} - příkaz pro definici způsobu výpisu čísla stránky (arabic, roman, atd.).

pagestyle{styl} - nastavení stylu stránkování (plain] číslo uprostřed dole, empty] žádné číslo, headings] standartní hlavičky, myheadings] vlastní hlavičky).

pagewidth - míra udávající šířku listu.

raggedbottom - deklarace, zamezuje zarovnání všech stránek na stejnou výšku, mezi odstavci jsou konstantní mezery.

voffset - délkový registr umožňující posunutí sazebního obrazce na stránce.

thispagestyle{styl} - příkaz nařizuje pro danou stránku příslušný styl číslování (plain, empty, headings, myheadings).

Seznam literatury

bibitem[**označení** {návěští}] - položka seznamu literatury v prostředí thebibliopgraphy, s definovaným označením a s návěštím, na něž se odvolává příslušný odkaz \cite.

cite[**text** {návěští}] - produkuje odkaz na citaci v textu. Návěští se odkazuje do seznamu literatury. Je-li uveden text, přidá se jako poznámka k odkazu.

nocite{seznam} - příkaz vyprodukuje do souboru .aux seznam citací, které lze poté zpracovat bibliografickou databází.

thebibliography - prostředí pro výčet citovaných publikací.

Skupina

{ - znak pro začátek skupiny.

} - znak pro konec skupiny.

Styly dokumentů] třídy, balíky

alltt - balík definující prostředí alltt.

article - standardní třída pro články, nejvyšší oddíl je section, stránkování plain.

book - standardní třída pro knihy, nejvyšší oddíl je charter, stránkování headings, dvoustránkový režim (twoside).

color - standardní balík definující práci s barvami.

graphics - standardní balík.

graphpap - standardní balík.

ifthen - standardní balík.

letter - standardní třída pro psaní dopisů.

pict2e - balík rozšiřující možnosti kreslení obrázků v prostředí picture.

report - standardní třída pro zprávy, nejvyšší oddíl je charter, stránkování plain.

slides - standardní třída pro tisk průhledných projekčních fólií. Dokument je složen z posloupnosti prostředí slide, definující obsahy jednotlivých fólií.

Prostředí tabular, array

& - oddělovač sloupců v prostředí tabular, eqnarray, array.

***-výraz** - výraz tvaru *{n}{def}, který se používá pro n-násobné opakování definice def v definici hlavičky tabulky, například `\*{4}{c|}` znamená totéž jako `\c|c|c|c|`.

<{deklarace} - v definici sloupců tabulky určuje materiál vkládaný za obsah každé buňky daného sloupce (balík array).

>{deklarace} - v definici sloupců tabulky určuje materiál vkládaný před obsah každé buňky daného sloupce (balík array).

-výraz - výraz v definici sloupců prostředí array a tabular definující prostor mezi sloupci. Například @{\ = \ } bude na každém řádku opakovat text " = ".

array - prostředí pro vytváření matic v matematickém režimu.

arraycolsep - délkový registr určující mezeru mezi sloupci v prostředí array.

arrayrulewidth - délkový registr určující tloušťku čáry použité v prostředí array a tabular.

arraystretch - koeficient určující mezeru mezi řádky prostředí array nebo tabular.

cline{odkud-kam} - prostředí array nebo tabular pro vodorovnou čáru počínaje sloupcem odkud až po sloupec kam včetně.

doublerulesep - míra udávající vzdálenost dvou sousedních vodorovných nebo svislých čar v prostředí array nebo tabular.

endfirsthead - definuje záhlaví tabulky prostředí longtable na první stránce. Definice je tvořena řádky, které předcházejí příkazu.

endhead - definuje záhlaví tabulky prostředí longtable na běžné stránce. Definice je tvořena řádky, které předcházejí příkazu.

endfoot - definuje zakončení tabulky prostředí longtable na běžné stránce. Definice je tvořena řádky, které předcházejí příkazu.

endlastfoot - definuje zakončení tabulky prostředí longtable na poslední stránce. Definice je tvořena řádky, které předcházejí příkazu.

extracolsep{míra} - příkaz užitý v @-výrazu, definující přídavnou mezeru velikosti míra mezi sloupci prostředí array, nebo tabular.

extrarowheight{míra} - délkový registr balíku array zvětšující svou velikostí výšku buněk tabulky.

hline - příkaz pro vodorovnou čáru v prostředí array nebo tabular.

jot - vertikální mezera přidávaná mezi řádky prostředí eqnarray nebo eqnarray*

kill - příkaz pro nevysázení řádku v prostředí longtable, uplatní se však jeho šířka.

lefteqn{vzorec} - příkaz, který představuje vzorec jako text nulové šířky. Tím umožní lepší zarovnání rovnic v prostředí eqnarray.

longtable[**pozice**]**{sldef}** - prostředí pro sazbu tabulek přesahujících jednu stránku. Horizontálně je tabulka umístěna podle volitelného parametru pozice, jehož hodnoty jsou l, r nebo c. Při jeho neuvedení se sází na střed. Sldef obsahuje sloupcovou definici jako u prostředí tabular.

LTcapwidth - šířka popisku prostředí longtable, implicitně 4in.

LTchunksize - počet řádků tabulky prostředí longtable, které má TeX zpracovávat najednou v paměti, implicitně je 20.

LTleft - pružná délka udávající levý okraj sazby tabulky prostředí longtable. Implicitně je `\fill`.

LTpost - pružná vertikální délka použitá za tabulkou prostředí longtable. Implicitně je `\bigskipamount`.

LTpre - pružná vertikální délka použitá před tabulkou prostředí longtable. Implicitně je `\bigskipamount`.

LTright - pružná délka udávající pravý okraj sazby tabulky prostředí longtable. Implicitně je `\fill`.

multicolumn{počet}{sl}{text} - příkaz nahrazující zápis obsahu počet, sloupců se způsobem zarovnání sl s textem text] prostředí array nebo tabular.

newcolumntype{p}{spec} - příkaz balíku array definující nový typ sloupce označeného p vytvořeného sloupcovou specifikací spec.

nonumber - příkaz zamezující na daném řádku prostředí eqnarray vytvoření čísla rovnice.

tabcolsep - délkový registr použitý k vodorovnému oddělené textu od okraje sloupce v prostředí tabular.

tabular - prostředí pro tvorbu tabulek a zarovnání textu ve sloupcích.

tabular* - prostředí pro tvorbu tabulek a zarovnání textu ve sloupcích s udanou celkovou šířkou.

tabularnewline[**míra**] - příkaz pro ukončení řádku tabulky. Míra udává vertikální vzdálenost k dalšímu řádku.

tabularx{sloupce} - prostředí pro tvorbu tabulek sázených na určitou šíři.

vline - příkaz pro vertikální čáru v prostředí array a tabular.

Prostředí tabbing

\' - definuje umístění textu na konec řádku.

\` - umísťuje text k zarážce zleva.

\+ - příkaz pro posun levého okraje o jednu zarážku vpravo.

- - příkaz pro posun levého okraje o jednu zarážku vlevo.

\< - přechod k předchozí tabelační zarážce.

\= - definuje pozici tabelační zarážky.

\> - přechod k následující tabelační zarážce.

a'{o} - akcent ó v prostředí tabbing.

a`{o} - akcent ò v prostředí tabbing.

a={o} - akcent ô.

kill - příkaz pro nevysázení řádku, uplatní se však nastavené zarážky.

poptabs - příkaz, nastavuje tabelační zarážky uložené dříve do paměťového zásobníku pomocí příkazu `\pushtabs`.

pushtabs - příkaz, který uschová současné nastavení tabelačních zarážek do paměťového zásobníku.

tabbing - prostředí pro zarovnání textu pod sebe] obdoba tabelačních zarážek na psacím stroji.

tabbingsep - délkový registr (parametr stylu) použitý při dosazení textu k dané tabelační zarážce příkazem `\'`.

Textová prostředí

center - prostředí pro sazbu textu na střed.

centering - příkaz pro sazbu na střed.

flushleft - prostředí pro sazbu na praporek vpravo.

flushright - prostředí pro sazbu na praporek vlevo.

quotation - prostředí používané pro delší citáty.

quote - prostředí používané pro krátké citáty.

raggedleft - příkaz pro sazbu na praporek vlevo.

raggedright - příkaz pro sazbu na praporek vpravo.

verse - prostředí pro sazbu veršů.

Tiskové zrcadlo

a4paper - sazba stránek velikosti A4 (210×297 mm).

a5paper - sazba stránek velikosti A5 (148×210 mm).

b5paper - volba pro sazbu stránek velikosti B5 (176×250 mm).

evensidemargin - míra určující nastavení levého okraje na stránce se sudým číslováním ve dvoustranném režimu.

footheight - míra udávající výšku boxu obsahujícího text paty stránky.

footskip - míra udávající vzdálenost posledního řádku od spodku paty stránky.

headheight - míra udávající výšku boxu obsahující hlavičku stránky.

headsep - míra udávající vzdálenost hlavičky od textu stránky.

executivepaper - sazba strányk rozměru 7,25×10,5 palce.

landscape - vzájemně zamění definovanou výšku a šířku stránky.

legalpaper -sazba stránek rozměru 8,5×14 palců.

letterpaper - sazba stránek rozměru 8,5×11 palců.

oddsidemargin - míra udávající velikost levého okraje na stránkách s lichými čísly (vpravo) ve dvoustranném režimu.

pageheight - míra, výška listu

pagewidth - míra, šířka listu.

textheight - míra, výška textového těla stránky.

textwidth - míra, šířka textového těla stránky.

topmargin - míra udávající velikost horního okraje stránky.

topskip - míra udávající vzdálenost hlavičky stránky od prvního řádku textu.

Titulní stránka

and - oddělení jmen v příkazu `\author`.

author{autoři} - příkaz pro vysazení jmen autorů na titulní stránce.

date{datum} - sazba data publice na titulní stránce.

maketitle - vytvoření titulní stránky dokumentu.

thanks{text} - vytvoří na titulní stránce poznámku text pod čarou.

title{titul} - příkaz pro vysázení titulu publikace na titulní stránce.

titlepage - prostředí produkující titulní stránku formátovanou podle přání uživatele. Titulní strana je nečíslovaná, následující je označena číslem 1.

Tři tečky

\ddots - $\ddots$

\dots - $\dots$

\ldots - $\ldots$

\vdots - $\vdots$

Velké oddělovače

leftodděl - příkaz pro vysazení velkého oddělovače.

rightodděl - příkaz, který způsobí vysázení oddělovače v takové velikosti, jaká je nutná pro vysázenou formuli.

Vložení souboru do zdrojového textu

\include{soubor} - příkaz pro zpracování souboru (dalšího zdrojového textu) v tom místě, kde je tento příkaz uveden.

\includeonly{seznam} - příkaz, který zamezí vložení všech souborů (příkazem `\include`), které nejsou uvedeny v seznamu.

\input{soubor} - příkaz pro zpracování souboru v místě, kde je tento příkaz uveden.

Vodorovná svorka

\overbrace{formule}^výraz - vytvoření vodorovné svorky orientované vzhůru. Svorka má šířku danou formulí (pouze v matematickém režimu). Ve vysazeném matematickém režimu lze ke špičce svorky umístit výraz.

\underbrace{formule}_výraz - vytvoření vodorovné svorky pod formulí. V matematickém prostředí lze umístit ke špičce svorky volitelně výraz.

Výčtová prostředí

description - prostředí pro výčet popisovaných položek.

enumerate - prostředí pro číslovaný výčet položek.

enumi . . . enumiv - čítače spřažené s různými úrovněmi vnoření prostředí enumerate a thebibliography.

\item - příkaz pro začátek položky ve výčtech itemize, enumerate, description, theindex.

\itemident - míra udávající odsazení návěští výčtu.

itemize - prostředí pro nečíslovaný výčet položek.

\itemsep - míra udávající vertikální vzdálenost mezi položkami výčtu.

labelitemi ... **labelitemiv** - příkazy produkující návěští položek výčtů itemize v závislosti na jejich vnoření.

labelsep - míra udávající vzdálenost návěští od levého okraje textu.

labelwidth - míra udávající šířku návěští výčtu.

leftmargin - míra udávající vzdálenost levého okraje výčtového prostředí od běžného levého okraje.

leftmargini ... - míry udávající velikost `\lefmargin` u vnořených výčtů.

list - prostředí pro obecný výčet.

listparindent - míra udávající ve výčtovém prostředí zarážku prvního řádku odstavce bez návěští.

makelabel{návěští} - příkaz generující návěští sázené pomocí příkazu `\item` ve výčtovém prostředí.

parsep - míra udávající vertikální vzdálenost odstavců ve výčtovém prostředí.

partopsep - míra, o kterou se zvyšuje vertikální odsazení výčtového prostředí od okolního textu.

rightmargin - míra udávající vzdálenost pravého okraje textu výčtového prostředí od běžného pravého okraje.

topsep - míra udávající svislou vzdálenost první položky výčtového prostředí od předcházejícího textu.

trivlist - prostředí pro jednoduchý výčet.

Výplně

dotfill - vodorovná výplň tečkami.

fill - pružná míra zabírající vždy maximální možný prostor.

hrulefill - vodorovná výplň souvislou čarou.

Zlomky

above míra - vysazení zlomku, jehož zlomková čára má tloušťku míra.

atop - vysazení dvou výrazů nad sebe.

brace - vysazení dvou výrazů nad sebe s okolními složenými závorkami.

brack - vysazení dvou výrazů nad sebe s okolními hranatými závorkami.

choose - vysazení kombinačního čísla.

frac{čitatel}{jmenovatel} - příkaz pro vysazení zlomku s vodorovnou zlomkovou čarou.

over - vytvoření zlomku s vodorovnou zlomkovou čarou. Objekty uvedené před tímto příkazem budou chápány jako čitatel, za příkazem budou chápány jako jmenovatel.

Dodatek C

Speciální znaky

V tomto dodatku jsou uvedeny všechny speciální znaky LaTeXu, které jsou intepretovatelné přímo do HTML kódu bez nutnosti využití skriptu `textogif`. V této verzi programu nemusely být využity, jsou zde ale uvedeny pro přídad, kdy by v příštích verzích programu využity být mohly.

U velkých písmen řecké abecedy (viz tabulka **??**.3), jsou uvedeny jen některé znaky, které se v matematickém prostředí LaTeXu liší od velkých písmen latinky. V HTML se tyto píšou podobně jako malé, jen s tím rozdílem, že první písmeno je velké (`&omega;` → `&Omega;`).

Tvary a šipky			
LaTeX příkaz symbolu	**vzhled symbolu**	**ekvivalent HTML**	**poznámka**
`\downarrow`	↓	`&darr;`	
`\Downarrow`	⇓	`&dArr;`	
`\leftrightarrow`	↔	`&harr;`	
`\Leftrightarrow`	⇔	`&hArr;`	
`\leftarrow`	←	`&larr;`	
`\Leftarrow`	⇐	`&lArr;`	
`\rightarrow`	→	`&rarr;`	
`\Rightarrow`	⇒	`&rArr;`	
`\uparrow`	↑	`&uarr;`	
`\Uparrow`	⇑	`&tuArr;`	
`\clubsuit`	♣	`&clubs;`	
`\diamondsuit`	♢	`&diams;`	
`\heartsuit`	♡	`&hearts;`	
`\spadesuit`	♠	`&spades;`	

Tabulka C.1: Tvary a šipky

Následují matematické a technické znaky. Tato tabulka samozřejmě neobsahuje všechny znaky saditelné LaTeXem. Obsahuje pouze znaky, které je možné napsat přímo v HTML kódu. Jejich vzhled se ale ve srovnání se znaky vysázenými LaTeXem se bude drobně lišit. To už je ale věc nesnadno ovlinitelná a proto je nutno spokojit se s takovýmto výsledkem.

Malé písmena řecké abecedy			
LaTeX příkaz symbolu	**vzhled symbolu**	**ekvivalent HTML**	**poznámka**
\alpha	α	α	
\beta	β	β	
\gamma	γ	γ	
\delta	δ	δ	
\epsilon	ϵ	ε	
\varepsilon	ε	ε	
\zeta	ζ	ζ	
\eta	η	η	
\theta	θ	θ	
\vartheta	ϑ	ϑ	
\iota	ι	ι	
\kappa	κ	κ	
\lambda	λ	λ	
\mu	μ	μ	
\nu	ν	ν	
\xi	ξ	ξ	
o	o	ο	
\pi	π	π	
\varpi	ϖ	ϒ	
\rho	ρ	ρ	
\varrho	ϱ		ekvivalent nenalezen
\sigma	σ	σ	
\varsigma	ς	ς	
\tau	τ	τ	
\upsilon	υ	υ	
\phi	ϕ	φ	
\varphi	φ		ekvivalent nenalezen
\chi	χ	χ	
\psi	ψ	ψ	
\omega	ω	ω	
Velké písmena řecké abecedy			
LaTeX příkaz symbolu	**vzhled symbolu**	**ekvivalent HTML**	**poznámka**
\Gamma	Γ	Γ	
\Delta	Δ	Δ	
\Theta	Θ	Θ	
\Lambda	Λ	Λ	
\Xi	Ξ	Ξ	
\Pi	Π	Π	
\Sigma	Σ	Σ	
\Upsilon	Υ	Υ	
\Phi	Φ	Φ	
\Psi	Ψ	Ψ	
\Omega	Ω	Ω	

Tabulka C.2: Řecká abeceda - velké a malé písmena

Matematické a technické znaky			
LaTeX příkaz symbolu	vzhled symbolu	ekvivalent HTML	poznámka
\angle	$\angle$	∠	
\approx	$\approx$	≈	
\cap	$\cap$	∩	
\cong	$\cong$	≅	
\cup	$\cup$	∪	
\emptyset	$\emptyset$	∅	
\equiv	$\equiv$	≡	
\exists	$\exists$	∃	
f	f	ƒ	
\forall	$\forall$	∀	
\infty	∞	∞	
\int	$\int$	∫	
\in	$\in$	∈	
\langle	$\langle$	⟨	
\lceil	$\lceil$	⌈	
\lfloor	$\lfloor$	⌊	
\ast	$\ast$	∗	
\nabla	∇	∇	
\neq	$\neq$	≠	
\ni	$\ni$	∋	
\notin	$\notin$	∉	
\not\subset	$\not\subset$	&notsub;	
\oplus	$\oplus$	⊕	
\vee	$\vee$	∨	
\otimes	$\otimes$	⊗	
\perp	$\perp$	⊥	
\pm	$\pm$	±	
\prod	$\prod$	∏	
\propto	$\propto$	∝	
\surd	$\surd$	√	
\rangle	$\rangle$	⟩	
\rceil	$\rceil$	⌉	
\rfloor	$\rfloor$	⌋	
\cdot	$\cdot$	⋅	
\sim	$\sim$	∼	
\subset	$\subset$	⊂	
\subseteq	$\subseteq$	⊆	
\sum	$\sum$	∑	
\supset	$\supset$	⊃	
\supseteq	$\supseteq$	⊇	

Tabulka C.3: Matematické a technické znaky