

Zpravodaj Československého sdružení uživatelů TeXu

Dominik Reháč

Priama sadzba dokumentov rôznych formátov v TeXu pomocou nástroja Pandoc

Zpravodaj Československého sdružení uživatelů TeXu, Vol. 31 (2021), No. 1-4, 83–92

Persistent URL: <http://dml.cz/dmlcz/150298>

Terms of use:

© Československé sdružení uživatelů TeXu, 2021

Institute of Mathematics of the Czech Academy of Sciences provides access to digitized documents strictly for personal use. Each copy of any part of this document must contain these *Terms of use*.



This document has been digitized, optimized for electronic delivery and stamped with digital signature within the project *DML-CZ*:
The Czech Digital Mathematics Library <http://dml.cz>

Priama sadzba dokumentov rôznych formátov v \TeX u pomocou nástroja Pandoc

DOMINIK REHÁK

\TeX ový balík Markdown umožňuje priamo sádzať dokumenty v jazyku Markdown a štylizovať jednotlivé prvky jazyka Markdown. Neponúka však podporu pre iné formáty dokumentov. Naopak program Pandoc umožňuje konverziu medzi desiatkami formátov dokumentov vrátane \TeX u a Markdownu, neumožňuje ale štylizovanie jednotlivých prvkov dokumentov.

Článok pojednáva o tom, ako je možné pridaním podpory medziformátů Pandocu do Markdownu umožniť priamu sadzbu rôznych textových formátov v \TeX ových dokumentoch. Zameriavam sa na množiny prvkov Markdownu a Pandocu a rozdiely medzi nimi, ktoré bude musieť nadchádzajúca implementácia prekonať. Nakoniec uvádzam plánované používateľské rozhranie pre \TeX .

Kľúčové slová: Markdown, Pandoc, Lua, \LaTeX

Úvod

Makrobalík *Markdown* [1] umožňuje doplniť dokumenty sádzané v \TeX u o úseky textu písané v odľahčenom značkovacom jazyku Markdown, ktoré počas sadzby transformuje na natívne \TeX ové makrá. Vzhľad dokumentov je možné ovplyvniť predefinovaním *medzimakier* sady `\markdownRenderer...`, ktoré Markdown interne využíva na vysádzanie jednotlivých prvkov textu. Makrobalík Markdown ale neponúka podporu pre iné značkovacie jazyky a formáty dokumentov.

Program *Pandoc* [2] umožňuje konverziu medzi desiatkami značkovacích jazykov a dokumentových formátov, ako sú dialekty Markdownu, formáty \TeX u (\LaTeX a \ConTeXt), HTML, docx, alebo roff. Výstup konverzie je možné ovplyvniť pomocou *filtrů*, čo sú používateľské programy, ktoré upravujú dokument počas konverzie. Pri konverzii do \TeX u má ale autor len obmedzené možnosti, ako ovplyvniť vzhľad výsledných dokumentov.

Integráciou Pandocu do balíku Markdown by bolo možné využiť silné stránky oboch. Konkrétne by takáto integrácia umožnila vysádzať v \TeX u dokumenty všetkých formátov, ktoré Pandoc podporuje ako vstupné, a upravovať vzhľad dokumentov jednoduchou úpravou medzimakier balíka Markdown. Medzimakrá balíka Markdown a prvky medziformátu programu Pandoc sa ale líšia a nie je možné ich priamočiaro na seba napojiť.

V tomto článku najprv zhŕňam aktuálne možnosti priamej sadzby rôznych formátov dokumentov v \TeX u. Následne popisujem medzimakrá balíka Markdown a medziformát programu Pandoc. Zameriavam sa na rozdiely medzi nimi

a navrhujem rozšírenie medzimakier Markdownu tak, aby lepšie korešpondovali s medziformátom Pandocu. Na záver ukazujem, ako bude používateľ $\text{T}_{\text{E}}\text{X}$ u môcť využiť Pandoc pre sadzbu rôznych formátov dokumentov.

Aktuálny stav vkladania dokumentov

Na vkladanie textových dokumentov iných formátov do $\text{T}_{\text{E}}\text{X}$ u momentálne existuje pár ďalších riešení, ktoré sa ale po bližšom preskúmaní ukážu ako nedostatočné.

Ak pre formát existuje možnosť exportu do PDF, je možné balíkom *pdfpages* [3] vložiť stránky vyexportovaného dokumentu priamo do dokumentu v $\text{T}_{\text{E}}\text{X}$ u. *pdfpages* ale neumožňuje žiadnu úpravu takto vložených stránok, čo nám znemožňuje akúkoľvek štylizáciu na strane $\text{T}_{\text{E}}\text{X}$ u. Taktiež tento prístup neumožňuje ani vloženie krátkych dokumentov do stredu súvislého textu, keďže *pdfpages* vie vkladať len celé stránky vloženého dokumentu pomedzi stránky $\text{T}_{\text{E}}\text{X}$ ového výstupu.

Existujú tiež rôzne riešenia špecializujúce sa na konkrétne vstupné formáty. Príkladom je *GrindEQ*, balík programov, ktorý okrem iného poskytuje konverziu dokumentov programu Microsoft Word priamo do $\text{T}_{\text{E}}\text{X}$ u.¹ Nemienim spochybňovať kvalitu takejto konverzie. *GrindEQ* je ale proprietárny softvér, vyžaduje zakúpenie licencie a beží vo vnútri programu Microsoft Word, ktorý je navyše viazaný na systém Microsoft Windows. Akákoľvek integrácia programov balíku *GrindEQ* priamo do $\text{T}_{\text{E}}\text{X}$ u by teda bola extrémne náročná až nemožná.

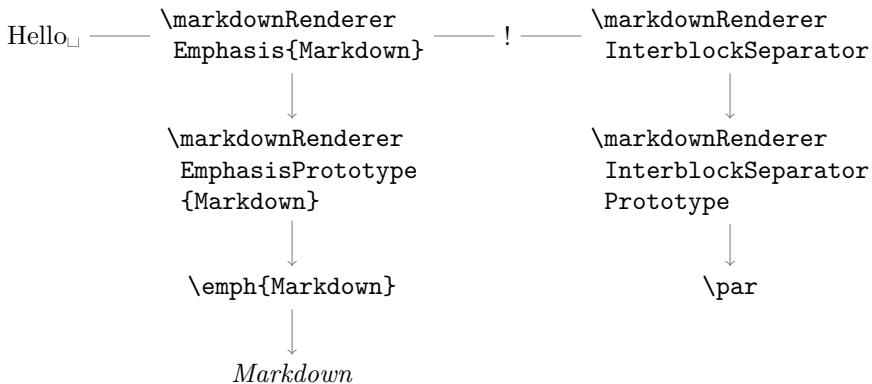
Popis architektúry Markdownu

Parser Markdownu v $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ovskom balíku Markdown pochádza z knižnice *lunamark* [4], za ktorou zhodou okolností tiež stojí John MacFarlane, autor Pandocu. Knižnica je zameraná na rýchlu konverziu Markdownu do iných bežne využívaných značkovacích formátov ako HTML alebo Docbook, ale i dvoch formátov $\text{T}_{\text{E}}\text{X}$ u: $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ u a $\text{C}\text{O}\text{N}\text{T}_{\text{E}}\text{X}\text{T}$ u. Keďže je *lunamark* napísaný v jazyku Lua, ktorý je možné priamo spúšťať v sádzacom systéme $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, javil sa parser *lunamarku* vhodný na jeho začlenenie do Markdownu.

Rýchlosť konverzie bola jednou z priorít pri vývoji *lunamarku*, a preto konverzia neprebíha v dvoch fázach (do medziformátu a z medziformátu) ako v prípade Pandocu. Namiesto toho tu dochádza k priamej komunikácii medzi dvomi modulmi, vstupným (reader) a výstupným (writer). Výstupný modul implementuje funkcie, ktoré definujú reprezentáciu rôznych prvkov textu vo výstupnom formáte a vstupný modul počas toho, ako parsuje vstupný dokument, zároveň aj zostavuje výstup s pomocou funkcií výstupného modulu, ktoré priamo volá.

Tento princíp bol zachovaný aj v *Markdowne* – s tým rozdielom, že existujúce výstupné moduly pre podporované formáty boli nahradené jedným, ktorý pre

¹<https://www.grindeq.com/index.php?p=word2latex>



Obr. 1: Reprézntácia medzimakier a výstupných makier balíku Markdown pre vstup „Hello *Markdown*!“

jednotlivé prvky textu generuje makrá `\markdownRenderer...` Tieto makrá je možné predefinovať a tak upraviť štýl výstupu. Predvolenou hodnotou týchto makier sú makrá `\markdownRenderer...Prototype`, ktoré expandujú na natívnu reprézntáciu príslušných prvkov systému $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ (viď obr. 1 a 2).

Popis architektúry Pandocu

Konverzia v Pandocu prebieha v dvoch fázach: konverziou vstupného formátu na natívnu reprézntáciu dokumentu a následnou konverziou tejto reprézntácie na výstupný formát. Dokumentácia Pandocu túto natívnu reprézntáciu nazýva *abstract syntax tree*, skrátene AST a vskutku sa jedná o strom prvkov Pandocu, ktorý sa uprostred konverzie celý nachádza v operačnej pamäti (viď obr. 3).

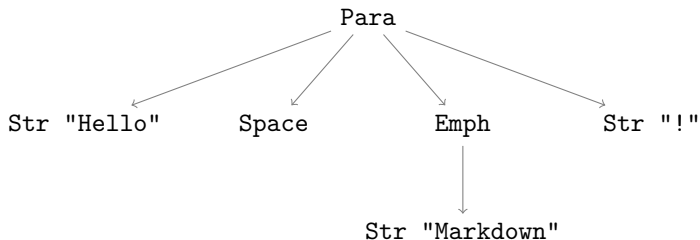
Pri zadaní výstupného formátu `native` vráti Pandoc na výstup celý tento strom vo svojej internej reprézntácii jazyka *Haskell*, v ktorom je Pandoc napísaný. Alternatívne je možné zadaním výstupného formátu `json` získať ekvivalentnú reprézntáciu tohoto stromu vo formáte JSON. Keďže existujú rôzne knižnice v jazyku Lua schopné parsovať formát JSON,² je pre naše rozšírenie Markdownu táto reprézntácia vhodnejšia.

Je nutné podotknúť, že Pandoc delí svoje prvky na blokové (block elements) a neblokové (inline elements). To ovplyvňuje, ktoré prvky sa môžu nachádzať v potomkoch jednotlivých prvkov a často korešponduje s výslednou podobou prvku, napr. implikuje medzery medzi odstavcami, ktoré sú blokovými prvkami. Pri spracovaní AST sa ale týmto rozdelením netreba veľmi zaoberať.

²Knižnica, ktorú naše rozšírenie využíva, je dostupná na <https://github.com/rxi/json.lua>.

1. `Tickbox Renderers`
 - (a) `TickedBox`
 - (b) `HalfTickedBox`
 - (c) `UntickedBox`
2. `InterblockSeparator`
3. `LineBreak`
4. `Ellipsis`
5. `Nbsp`
6. `Special Character Renderers`
 - (a) `Ampersand`
 - (b) `Backslash`
 - (c) `Circumflex`
 - (d) `DollarSign`
 - (e) `Hash`
 - (f) `LeftBrace`
 - (g) `PercentSign`
 - (h) `Pipe`
 - (i) `RightBrace`
 - (j) `Tilde`
 - (k) `Underscore`
7. `CodeSpan`
8. `Link`
9. `Image`
10. `ContentBlock`
11. `Bullet List`
 - (a) `UlBegin`
 - (b) `UlBeginTight`
 - (c) `UlItem`
 - (d) `UlItemEnd`
 - (e) `UlEnd`
 - (f) `UlEndTight`
12. `Ordered List`
 - (a) `OlBegin`
 - (b) `OlBeginTight`
 - (c) `OlItem`
 - (d) `OlItemEnd`
 - (e) `OlItemWithNumber`
 - (f) `OlEnd`
 - (g) `OlEndTight`
13. `Definition List`
 - (a) `DlBegin`
 - (b) `DlBeginTight`
 - (c) `DlItem`
 - (d) `DlItemEnd`
 - (e) `DlDefinitionBegin`
 - (f) `DlDefinitionEnd`
 - (g) `DlEnd`
 - (h) `DlEndTight`
14. `Emphasis`
 - (a) `Emphasis`
 - (b) `StrongEmphasis`
15. `Block Quote`
 - (a) `BlockQuoteBegin`
 - (b) `BlockQuoteEnd`
16. `Code Block`
 - (a) `InputVerbatim`
 - (b) `InputFencedCode`
17. `YAML Metadata`
 - (a) `JekyllDataBegin`
 - (b) `JekyllDataEnd`
 - (c) `JekyllDataMappingBegin`
 - (d) `JekyllDataMappingEnd`
 - (e) `JekyllDataSequenceBegin`
 - (f) `JekyllDataSequenceEnd`
 - (g) `JekyllDataBoolean`
 - (h) `JekyllDataNumber`
 - (i) `JekyllDataString`
 - (j) `JekyllDataEmpty`
18. `Heading`
 - (a) `HeadingOne`
 - (b) `HeadingTwo`
 - (c) `HeadingThree`
 - (d) `HeadingFour`
 - (e) `HeadingFive`
 - (f) `HeadingSix`
19. `HorizontalRule`
20. `Footnote`
21. `Cite`
22. `TextCite`
23. `Table`
24. `InlineHtmlComment`

Obr. 2: Úplný zoznam výstupných makier balíku Markdown k verzii 2.11.0



Obr. 3: Pandoc AST pre vstup Markdownu „Hello *Markdown*!“

- | | |
|--------------------|-----------------|
| 1. Plain | 18. Strong |
| 2. Para | 19. Strikeout |
| 3. LineBlock | 20. Superscript |
| 4. CodeBlock | 21. Subscript |
| 5. RawBlock | 22. SmallCaps |
| 6. BlockQuote | 23. Quoted |
| 7. OrderedList | 24. Cite |
| 8. BulletList | 25. Code |
| 9. DefinitionList | 26. Space |
| 10. Header | 27. SoftBreak |
| 11. HorizontalRule | 28. LineBreak |
| 12. Table | 29. Math |
| 13. Div | 30. RawInline |
| 14. Null | 31. Link |
| 15. Str | 32. Image |
| 16. Emph | 33. Note |
| 17. Underline | 34. Span |

Obr. 4: Úplný zoznam prvkov medziformátu programu Pandoc

Podobnosti a rozdiely medzi architektúrami

Integrácia programu Pandoc do nástroja Markdown vyžaduje jednoznačné mapovanie z medziformátu Pandocu do medzimakier Markdownu. V tomto oddiele sa najprv zameriavam na prvky Pandocu, ktoré majú svoje ekvivalenty v makrách Markdownu. Následne popisujem prvky unikátne pre Pandoc a navrhujem rozšírenia makier Markdownu, ktoré zaistia bezproblémovú konverziu.

Spoločné prvky

Zopár prvkov Pandocu už dokonale korešponduje s existujúcimi makrami Markdownu a ich transformácia je triviálna. Jedná sa napríklad o zvýraznenia **Emph** a **Strong**, ktorým zodpovedajú makrá **Emphasis** a **StrongEmphasis**.³ Tieto prvky majú jeden parameter, rovnako ako uvedené makrá, takže naše spracovanie by končilo volaním príslušných prototypov makier nad daným parametrom. Do tejto skupiny by sme mohli zaradiť aj prvky **HorizontalRule** a **LineBreak** s rovnomenými makrami, ako i prvky **Note**, **Code** a **CodeBlock** – tento síce s opačným poradím parametrov ako makro – s rešpektívnymi makrami **Footnote**, **CodeSpan** a **InputFencedCode**.

Iné prvky je nutné spracovať skôr, než ich dokážeme prepísať na existujúce makrá. Príkladom je prvok **Header**, ktorý si okrem svojho obsahu nesie i celé číslo definujúce, o kolkú úroveň hlavičky sa jedná. Tomu zodpovedajú makrá **HeadingOne** až **HeadingSix**. Podobným prípadom sú makrá zoznamov **OrderedList**, **BulletList** a **DefinitionList**, ktoré pozostávajú z polí vyjadrujúcich jednotlivé položky. Ich ekvivalentmi v Markdowne sú makrá vyznačujúce začiatky a konce celých štruktúr, ako aj ich položiek – napr. **UlBegin**, **UlEnd**, **UlItemBegin** a **UlItemEnd** u neusporiadaných zoznamov. (To isté platí i pre prvok **BlockQuote** a dvojicu makier **BlockQuoteBegin** + **BlockQuoteEnd**.)

Niekedy je toto spracovanie prvkov menej komplikované. Príkladom je makro **Cite**, ktoré môže v závislosti na počte poskytnutých citácií prijať rôzny počet parametrov, pričom prvým z nich je počet citácií. Získanie tohoto počtu z príslušného prvku **Cite** je v jazyku Lua triviálne, ostatné parametre potom získame určitým usporiadaním obsahov citácií.

Odkazy a obrázky Pandoc vyjadruje prvkami **Link** a **Image**. Prvým parametrom **Link** je zoznam prvkov tvoriaci text odkazu a druhým je dvojica (URL, titulok) špecifikujúca odkaz samotný.⁴ Oproti tomu **Title** má parametre štyri – okrem textu odkazu a titulku prijíma dve URI, z nich jedno v nezmenenej podobe a druhé vhodné priamo na sadzbu, t. j. s ošetrovanými špeciálnymi znakmi. Do dokumentu sa reálne dostane len ošetrovaná URI, využitie nezmenenej je na použi-

³Z dôvodu úspory miesta v tomto oddiele uvádzam skrátené názvy makier, t. j. napr. **Emphasis** a **StrongEmphasis** miesto `\markdownRendererEmphasis` a `\markdownRendererStrongEmphasis`.

⁴Titulok zodpovedá hodnote nepovinného HTML atribútu `title`.

teľovi. Podobnú formu má prvok `Image` a jeho rovnomenné makro, kde je naopak predvolene využitá len nezmenená URI, ktorá určuje cestu k obrázku.

Je nutné podotknúť, že u niektorých prvkov by využitie existujúcich makier mohlo byť mierne stratové. Napríklad u prvku `OrderedList` je možný aj výskyt dvoch vyčísľiteľných typov, ktoré špecifikujú štýl zobrazenia čísel položiek⁵ a oddeľovače týchto čísel⁶. To ale nedokážeme vyjadriť existujúcou skupinou makier `OlBegin`. . . Tieto makrá tiež nemôžeme predefinovať, pretože by to mohlo viesť k spätnej nekompatibilitate. (Toto je síce nad rámec môjho počiatočného rozšírenia, keďže ide o zriedkavo vyskytujúce sa prvky, ale do budúcnosti nevyklúčujem implementáciu úplnej reprezentácie cez prídavné makrá.)

Podobná strata by nastala pri spracovaní prvku `Table`, štruktúrne najzložitejšieho prvku `Pandocu`, príslušným makrom. Prvým parametrom `Table` je titulok tabuľky, za ktorým nasleduje pole dvojíc vyčísľiteľných typov `Alignment` a `ColWidth`. Každá dvojica špecifikuje zarovnanie jedného zo stĺpcov, ako aj jeho relatívnu šírku. Zvyšné tri parametre tvoria záhlavie, zoznam tiel a päť tabuľky. (Každé telo je navyše dvojica zložená z nepovinného záhlavia a hlavnej časti tela.) Všetky tieto časti tabuľky pozostávajú z niekoľkých zoznamov, ktoré vyjadrujú rady jednotlivých polí tabuľky.

Štruktúra makra `Table` je o niečo jednoduchšia a viac plochá. Prvými tromi parametrami sú titulok a dve prirodzené čísla R a S , ktoré určujú počet riadkov a stĺpcov tabuľky. Nasleduje reťazec S znakov `d`, `l`, `c` alebo `r`, ktorý určuje smer zarovnania v jednotlivých stĺpcoch. Na záver makro prijme R parametrov, každý z nich zložený z S parametrov, ktoré opäť vyjadrujú jednotlivé polia tabuľky. Vyjadriť šírky stĺpcov a určiť, ktoré riadky sú záhlavia, makro `Table` už nevie.

Prvky unikátne pre Pandoc

Tým sa dostávame k druhej polovici prvkov, ktorá nemá v makrách `Markdownu` svoje ekvivalenty. Patria sem napríklad jednoduché prvky `Underline`, `Strikeout`, `Superscript`, `Subscript` a `SmallCaps`, ktorých význam je zrejmý. Implementovať tieto makrá bude triviálne, ak vieme, ako rovnakého efektu docieľiť v `TeXu`.

O niečo zaujímavejší je napríklad prvok `Quoted`, ktorý vyjadruje úsek textu ohraničený úvodzovkami. Okrem poľa potomkov, ktoré ohraničuje, obsahuje `Quoted` ešte hodnotu vyčísľiteľného typu `QuoteType`, ktorý nadobúda hodnotu `SingleQuote` alebo `DoubleQuote` a tá vyjadruje, či sa jedná o jednoduché alebo dvojité úvodzovky. Po vzore makier `Heading`. . . by bolo ideálne toto na strane `Markdownu` vyjadriť dvomi samostatnými makrami, čím sa zachová konzistencia balíku. Podobne prvok `Math` obsahuje vyčísľiteľný typ `MathType` s hodnotou `InlineMath` alebo `DisplayMath`, ktorý určuje, či sa majú priložené literály vysádzať v rámci riadku alebo ako samostatná rovnica.

⁵desiatkové čísla, veľké/malé rímske číslice alebo veľké/malé písmená

⁶bodka, zátvorka alebo dve zátvorky

Zvláštne prípady tvoria prvky **Str** a **Plain**, ktoré vyjadrujú „čistý“ úsek textu bez dodatočného značkovania⁷ a tým pádom by bolo možné priamo vysádzať ich obsah. To isté môžeme povedať o prvku **Space**, ktorý reprezentuje jedinú medzeru alebo **Null**, ktorý je prázdny a nevyjadruje nič. Všetky tieto prvky by bolo triviálne nahradiť – v prípade **Null** dokonca zahodiť. Je ale možné, že by niektorí z používateľov Markdownu pre ne dokázali nájsť zmysluplné využitie. Preto pre ne tiež zavedieme príslušné makrá.

Na členenie textu poskytuje Pandoc prvky **Para**, **LineBlock** a **SoftBreak**. **Para** je zhruba ekvivalentný novému riadku s odsadením, **SoftBreak** novému riadku bez odsadenia. **LineBlock** potom vyjadruje skupinu neodsadených riadkov.

RawBlock a **RawInline** sú prvky označujúce úseky textu v iných formátoch. Generuje ich napríklad HTML značkovanie využité vo formáte Markdown. V dobe písania tohto článku stále nie je úplne jasné, ako s týmito prvkami nakladať. Ak by šlo o jeden z **TeX**ových formátov, bolo by možné text jednoducho vložiť priamo do dokumentu, čo ale so sebou môže niesť isté bezpečnostné riziko. Ponúka sa tiež možnosť opakovanej konverzie týchto blokov na AST, čo by ale z pochopiteľných dôvodov bolo možné, len ak je Pandoc dostupný pri sadzbe, čo nemusí, ak len podsúvame Markdownu inak získaný AST.

Takisto nie je jasné, ako spracovať prvky **Div** a **Span**. Tie zodpovedajú HTML značkám `<div>` a `` a ich príslušným nepovinným atribútom `label` a `class`, ktoré ale môžu obsahovať prakticky čokoľvek. Najrozumnejšou cestou asi bude predvolene vysádzať obsah týchto prvkov bez akýchkoľvek úprav a prípadné prvky so špeciálnymi atribútmi nechať na používateľa.

Makrá ako **TickedBox** alebo **Ellipsis**, ktoré som v tomto oddiele nespomenul, nemajú v Pandocu svoje ekvivalenty. To je, čo sa tohto rozšírenia týka, v poriadku.

Špecifiká plánovaného rozšírenia

Čo sa integrácie samotnej týka, najjednoduchšou cestou je pravdepodobne rozšíriť lunamark o vstupný modul schopný spracovania AST Pandocu a tento model následne integrovať do Markdownu. V dobe písania článku je práca na takomto module takmer hotová.⁸

Pandoc takisto umožňuje používateľom poskytnúť vlastný výstupný modul v jazyku Lua⁹, takže alternatívnym riešením by bolo napísať výstupný modul, ktorý by priamo generoval makrá Markdownu. To by ale vyžadovalo prítomnosť nainštalovaného Pandocu na počítači, kde prebieha sadzba.

Spôsoby, ktorými bude rozšírenie možné využiť v dokumentoch, budú pravdepodobne kopírovať existujúce prostriedky na sadzbu jazyka Markdown. Na priamu

⁷Rozdiel **Str** a **Plain** spočíva v tom, že **Plain** je blokový a **Str** nie je. **Plain** sa vyskytuje okrem iného napr. v poliach tabuliek. Príklad výskytu **Str** je uvedený na obr. 1.

⁸https://github.com/drehak/lunamark/blob/devel/lunamark/reader/pandoc_json.lua

⁹<https://pandoc.org/MANUAL.html#custom-readers-and-writers>

```

\documentclass{article}
\usepackage{markdown}
\begin{document}
\begin{pandoc}{format=man}
.TH "Pandoc User\{cq}s Guide" "" "July 18, 2021" "pandoc 2.14.1" ""
.hy
.SH NAME
pandoc - general markup converter
.SH SYNOPSIS
.PP
\f[C]pandoc\f[R] [\f[I]options\f[R]] [\f[I]input-file\f[R]]\&...
.SH DESCRIPTION
.PP
Pandoc is a Haskell library for converting from one markup format to
another, and a command-line tool that uses this library.
\end{pandoc}
\end{document}

```

Obr. 5: Ukážka využitia prostredia pandoc

NAME

pandoc - general markup converter

SYNOPSIS

pandoc [*options*] [*input-file*]...

DESCRIPTION

Pandoc is a Haskell library for converting from one markup format to another, and a command-line tool that uses this library.

Obr. 6: Ukážka výstupu prostredia pandoc

```

\documentclass{article}
\usepackage{markdown}
\begin{document}
\pandocInput[format=docx]{documentation-of-pandoc.docx}
\end{document}

```

Obr. 7: Ukážka využitia makra pandocInput

sadzbu bude podľa vzoru prostredia `markdown` dostupné prostredie `pandoc`, ktoré ale bude navyše vyžadovať informáciu o tom, o aký vstupný formát sa jedná (viď obr. 5 a 6). (Konečná forma sa môže mierne líšiť od ukážky.)

Niektoré vstupné formáty Pandocu, napr. *docx*, ale majú binárnu podobu, ktorá nie je vhodná na vkladanie do $\text{T}_{\text{E}}\text{X}$ ovských dokumentov, a preto pre ich sadzbu bude nutné využiť makro `\pandocInput`, analogické k existujúcim makrám `\markdownInput` a `\input` (viď obr. 7).

Referencie

1. NOVOTNÝ, Vít. Sazba textu označovaného v jazyce Markdown uvnitř $\text{T}_{\text{E}}\text{X}$ ových dokumentů. *Zpravodaj Československého sdružení uživatelů $\text{T}_{\text{E}}\text{X}$* . 2016, č. 1–4, s. 78–93. Dostupné z DOI: 10.5300/2016-1-4/78.
2. MACFARLANE, John. *Pandoc: a universal document converter* [online]. 2006 [cit. 2021-11-22]. Dostupné z: <https://pandoc.org/>.
3. MATTHIAS, Andreas. *The pdfpages Package* [online]. 2021-03-06 [cit. 2021-11-22]. Dostupné z: <https://ctan.org/pkg/pdfpages>.
4. MACFARLANE, John; HAGEN, Hans; HOSNY, Khaled. *Lunamark* [online]. 2009 [cit. 2021-11-22]. Dostupné z: <http://jgm.github.io/lunamark/>.

Summary: Direct Typesetting of Various Document Formats in $\text{T}_{\text{E}}\text{X}$ Using the Pandoc Utility

The Markdown $\text{T}_{\text{E}}\text{X}$ package allows authors to typeset documents in the Markdown language and maintain control over how the documents will look. However, the package doesn't provide support for document formats other than Markdown. In contrast, the Pandoc tool enables the conversion between dozens of document formats including $\text{T}_{\text{E}}\text{X}$ and Markdown, but only provides rudimentary control over styling.

This article elaborates on the possibility of typesetting various text formats directly in $\text{T}_{\text{E}}\text{X}$ by adding support for Pandoc's intermediate document representation into the Markdown package. I focus mainly on the intermediate representations of Markdown and Pandoc as well as the differences between them, which my upcoming implementation will have to overcome. At the end of my article, I present the planned user interface for $\text{T}_{\text{E}}\text{X}$.

Keywords: Markdown, Pandoc, Lua, $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$

Dominik Reháč, drehak@firemail.cc