

Mendelova univerzita v Brně
Provozně ekonomická fakulta

Moderní přístupy k tvorbě tabulek v systému T_EX/L^AT_EX

Diplomová práce

Vedoucí práce:
Ing. Jan Přichystal, Ph.D.

Bc. Jan Prokeš

Brno, 2013

Děkuji Ing. Janu Přichystalovi, Ph.D. za vedení této diplomové práce a doc.
Ing. Dr. Jiřímu Rybičkovi za poskytnutí konzultací během tvorby této práce.

Prohlašuji, že jsem tuto práci vytvořil samostatně a s použitím literatury uvedené v kapitole Použitá literatura.

V Brně 23. května 2013

.....

Abstract

Prokeš, J. Modern approaches to a table creation in the $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ system. Diploma thesis. Brno, 2013.

This diploma thesis concerns a table creation in the $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ system. An analysis of tables in some existing documents and resulting definitions of requirements to a style is its part. Main part of this thesis is dedicated to a creation of the style for tables typesetting, which should realize mentioned requirements.

Keywords

table, $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{T}_{\text{E}}\text{XonWeb}$

Abstrakt

Prokeš, J. Moderní přístupy k tvorbě tabulek v systému $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Diplomová práce. Brno, 2013.

Tato diplomová práce se zabývá problematikou tvorby tabulek v typografickém systému $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Její součástí je analýza tabulek v některých existujících dokumentech a následná definice požadavků na styl. Hlavní část této práce je věnována tvorbě stylu pro sazbu tabulek, který by měl zmíněné požadavky splnit.

Klíčová slova

tabulka, $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, $\text{T}_{\text{E}}\text{XonWeb}$

Obsah

1	Úvod a cíl práce	8
1.1	Úvod	8
1.2	Cíl práce	8
2	Metodika práce	9
3	Tabulky, jejich typografická pravidla a různé přístupy k tvorbě tabulek v některých programových nástrojích	10
3.1	Pořadová sazba	10
3.2	Tabulky	11
3.2.1	Typy tabulek	14
3.3	Diagramy	16
3.4	Prostředí pro sazbu tabulek v typografickém systému T _E X/L ^A T _E X . . .	16
3.4.1	Prostředí <code>tabbing</code>	17
3.4.2	Prostředí <code>tabular</code>	18
3.4.3	Balík <code>colortbl</code>	20
3.5	Přístup k tvorbě tabulek v jazyce HTML	20
3.6	Přístup k tvorbě tabulek v textovém procesoru Microsoft Word	21
4	Analýza existujících dokumentů obsahujících tabulky a definice požadavků	23
4.1	Analýza existujících dokumentů	23
4.1.1	Sledované vlastnosti a formátování	23
4.1.2	Výsledky analýzy	24
4.2	Definice požadavků	25
5	Výběr implementačních nástrojů	27
5.1	Boxy	27
5.1.1	LR boxy	28
5.1.2	Parboxy	28
5.1.3	Rule boxy	28
5.2	Volba nástroje pro sazbu grafických prvků	28
5.2.1	Prostředí <code>picture</code>	28
5.2.2	Balík <code>color</code>	29
5.2.3	Balíky PGF a TikZ	29
6	Implementace	30
6.1	Soubory stylu	30
6.2	Základy stylu	30
6.2.1	Příkazy definující parametry	31
6.2.2	Změna výchozího nastavení pro všechny tabulky v dokumentu	32
6.2.3	Změna výchozího nastavení pro určitou tabulku v dokumentu	33

6.2.4	Změna výchozího nastavení pro část tabulky	33
6.2.5	Definice sloupců, meziřádkových mezer a příp. uvedení titulku tabulky	35
6.2.6	Definice buněk	36
6.2.7	Definované délkové registry a jejich využití	37
6.2.8	Definované čítače a jejich využití	37
6.2.9	Definované struktury a jejich využití	38
6.3	Prostředí <code>tableenv</code>	39
6.4	Používání vrstev	40
6.5	Slovně pojmenované délky	42
6.6	Sloupec a mezisloupcová mezera	43
6.6.1	Sloupec	43
6.6.2	Mezisloupcová mezera	44
6.7	Řádek a meziřádková mezera	45
6.7.1	Ukončení řádku	45
6.8	Výchozí stupeň písma v různých prvcích tabulky	46
6.9	Titulek tabulky	47
6.9.1	Implementace příkazu	48
6.10	Poznámky pod tabulkou	49
6.10.1	Příkaz <code>\tablenote</code>	50
6.10.2	Příkaz <code>\tablenotenum</code>	50
6.10.3	Příkaz <code>\tablemark</code>	50
6.10.4	Sazba poznámek pod tabulkou, její formátování a implementace	51
6.11	Buňka	52
6.11.1	Sloučená buňka	52
6.11.2	Zjištění výšky řádku	53
6.11.3	Zjištění šířky buňky	53
6.11.4	Obsah buňky a mezery od okrajů buňky	53
6.11.5	Linky okolo buňky	54
6.11.6	Pozadí buňky	58
6.11.7	Rámeček buňky	58
6.11.8	Přeškrtnutí buňky	59
6.11.9	Shrnutí sazby buněk	59
6.12	Tvorba pořadové sazby, různých typů tabulek a diagramů v prostředí <code>\tableenv</code>	60
6.12.1	Tvorba pořadové sazby	60
6.12.2	Tvorba tabulek	61
6.12.3	Tvorba diagramů	63
6.12.4	Nasazení stylu v projektu <code>TeXonWeb</code>	64
7	Zhodnocení	65
7.1	Omezení stylu	66
7.2	Práce se stylem	67

OBSAH	7
8 Závěr	68
9 Použitá literatura	69
Přílohy	70
A Seznam dokumentů použitých v analýze	71
B CD příloha	74

1 Úvod a cíl práce

1.1 Úvod

Tabulky jsou velmi výrazné prvky dokumentů, které čtenáře zaujmou. Údaje seskupené do tabulky jsou přehledně uspořádané a čtenář se v nich snadno orientuje. To také usnadňuje jejich zapamatovatelnost. Proto se tabulky využívají ke sdělování informací, jimž je třeba věnovat zvláštní pozornost. Tabulky tedy mají velký význam a odborné publikace se bez nich neobejdou. (Talandová, 2006)

Sazbu tabulek umožňuje velké množství programového vybavení. Různá programová vybavení přistupují k tvorbě tabulek různými způsoby. Naprosto odlišně se např. vytváří tabulka v textovém procesoru Microsoft Word a v jazyce HTML. Tyto přístupy k tvorbě tabulek jsou v práci zmíněny, ale tato práce se však zaměřuje především na problematiku sazby tabulek v typografickém systému $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$. V tomto typografickém systému existuje pro sazbu tabulek velké množství prostředí a balíků. Přesto ale dosud nelze říci, že by sazba tabulek v $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ byla vyřešeným problémem. Existujícími prostředími a balíky pro sazbu tabulek se mj. velmi podrobně zabývá (Talandová, 2006) a to také včetně jejich vzájemné kompatibility.

Úkolem této práce je také vytvořit nový styl pro sazbu tabulek. Při jeho implementaci nebude použito žádné prostředí určené primárně pro tvorbu tabulek. Může tak být také odhaleno mnoho problémů, které zahrnuje tvorba nového prostředí pro sazbu tabulek, což může být rovněž přínosem této práce. Počítá se s tím, že vytvořený styl bude dostupný v projektu $\text{T}_{\text{E}}\text{XonWeb}$ (Přichystal, 2013).

1.2 Cíl práce

Cílem této diplomové práce je uživatelům typografického systému $\text{T}_{\text{E}}\text{X}/\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ umožnit co nejkomplexnější sazbu tabulek prostřednictvím nového stylu. Základním požadavkem na tento styl je zahrnout běžně nejpoužívanější formátování tabulek. Je také podstatné umožnit co nejjednodušší budoucí rozšíření stylu.

2 Metodika práce

Jako první krok v této práci je nutné zaměřit se na literaturu věnující se tabulkám. V nich je třeba vyhledat informace o tom, jak se tabulky správně vytváří, jak se tabulky dělí a další teoretické poznatky, protože mnohé z nich bude potřeba ve stylu přímo zohlednit.

Poté je třeba uvést, jak ke tvorbě tabulek přistupují různá programová vybavení a největší pozornost věnovat typografickému systému $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. V něm se pro sazbu tabulek využívá prostředí a balíků. Je nutné uvést přehled alespoň těch nejpoužívanějších. Vhodné je také ukázat na jednoduchém příkladu jejich používání.

Aby bylo možné definovat požadavky na styl, je potřeba analyzovat velké množství tabulek v různých již existujících dokumentech a zjistit, která formátování tabulek v nich jsou použita a jak často. Teprve na základě této analýzy bude možné vytvořit definici požadavků na styl pro sazbu tabulek. Bude kladen velký důraz na to, aby nejvíce použité možnosti formátování byly zahrnuty do implementace stylu. Méně používaná formátování budou mít nižší prioritu.

V dalším kroku bude potřeba provést výběr implementačních prostředků, které budou během implementace využity. To zahrnuje především výběr vhodných prostředí, balíků a dalších nástrojů, které umožní splnit definované požadavky.

Poté již bude možné styl implementovat. Cílem bude dosáhnout co nejvyšší komplexnosti stylu, především v něm ale musí být dosaženo implementace již zmíněných nejčastěji používaných možností formátování. V této práci bude, samozřejmě kromě popisu struktury a způsobu používání stylu, seznámení s definovanými příkazy, atd., také velmi podrobně popsán způsob implementace jednotlivých prvků. To může uživatelům velmi pomoci nejen při provádění drobných změn ve stylu, ale např. také při rozšiřování stylu o v něm dosud neimplementovanou funkcionalitu.

Po implementaci bude vytvořený styl nasazen v již zmíněném projektu $\text{T}_{\text{E}}\text{X}$ onWeb.

Následovat již bude zhodnocení vytvořeného stylu. Budou uvedeny jeho nejvýznamnější přednosti a nedostatky. Budou také uvedeny návrhy na jeho příp. budoucí rozšíření, které by styl dále vylepšily, např. zvýšily komplexnost stylu.

3 Tabulky, jejich typografická pravidla a různé přístupy k tvorbě tabulek v některých programových nástrojích

3.1 Pořadová sazba

V pořadové sazbě je text uspořádán do přehledného pořadí pod sebou v samostatných řádkách nebo do sloupců vedle sebe. Takto se sázejí knižní obsahy, jmenné, místní, pojmové seznamy, statistické přehledy, jména, předměty, míry, číselné hodnoty a jiné. (Pop, Flégr, Pop, 1989)

Jednodušší textové skupiny se obvykle sázejí zkraje, pokud text přechází do druhé řádky, popřípadě dalších řádek, pro přehlednost a lepší estetický vzhled se tyto řádky odrážejí čtverčíkem. (Pop, Flégr, Pop, 1989)

Méně užívaný způsob jsou pořadově sázené texty se zarážkou u první řádky. Rozhodující je, aby zarážky byly vždy v určitém rytmu pravidelně pod sebou. (Pop, Flégr, Pop, 1989)

V sazbě knižních rejstříků, slovníků i jinde se k opakování základních významů obvykle používá pomlčka. (Pop, Flégr, Pop, 1989)

Ve složitějších textech sázených pořadově se řádky významově důležitější odlišují od významově méně důležitých textových skupin označením na začátku řádky nebo odpovídající řádkovou zarážkou. K označení se používají římské číslice s tečkou, verzálky s tečkou, arabské číslice s tečkou nebo minuskule se závorkou. (Pop, Flégr, Pop, 1989)

Zásadně se nesázejí tečky a závorky dohromady. Ve sloupcích se podřazené znaky vyrovnávají podle teček, resp. závorek. (Pop, Flégr, Pop, 1989)

Při kombinaci arabské číslice s minuskulí se neklade tečka za číslo, ale jen závorka za minuskulí. (Pop, Flégr, Pop, 1989)

Arabské číslice se sázejí s tečkou, jsou odlévány na půlčtverčík a také tečka je odlévána na systém, tj. na čtvrtinu čtverčíku. U druhé řádky se potom snadno tvoří odpovídající textová zarážka pomocí systematického písmového výplňku a vždy je záruka, že číslice i text se budou krýt pod sebou. Arabské číslice se musí stavět pod sebe podle hodnoty (jednotky, desítky, sta, tisíce atd.). (Pop, Flégr, Pop, 1989)

Římské číslice se používají pro označení století, měsíců, kapitol v knihách apod. Sázejí se z písmových verzálek a jejich šíře je tedy nestejná a nesystematická. Sázejí se s tečkou, pouze při označování jednotlivých kapitol v knize se sázejí bez tečky. Jsou-li v pořadové sazbě pod sebou, vysadí se nejdříve číslo nejširší a ostatní se vyplní podle něho doprava tak, aby tečky přišly pod sebe. Kombinací jednotlivých verzálkových liter v určitém pořadí je možné sestavit i čísla vyšších hodnot. Římské číslice se píší a čtou v pořadí tak, že hodnoty psané vpravo od nejvyšší hodnoty se připočítávají a psané vlevo se od ní odečítají. (Pop, Flégr, Pop, 1989)

Některá čísla lze sestavit z různých sestav, které jsou hodnotově stejné. Správná je kratší sestava, složená z méně znaků. (Pop, Flégr, Pop, 1989)

Při vyznačení odstavce verzálkami se postupuje obdobně jako u sazby z římských čísel, tzn. že se podle nejširší verzálky zpředu doplní ostatní tak, aby tečky byly pod sebou. (Pop, Flégr, Pop, 1989)

Při vyznačování písmeny malé abecedy doplňují se jednotlivá písmena kulatou závorkou. Písmena mohou být sázena z kurzívy pro zvýraznění odlišení od základního textu. Také závorky stejně jako tečky u verzálek a číselného označení se sázejí pod sebou. (Pop, Flégr, Pop, 1989)

Desetinné třídění je přehledný a jednoduchý způsob uspořádání pořadové sazby, uplatňovaný zejména v sazbě odborné literatury, kde je nutné podrobné členění. Pro označení podřazenosti se zde používají arabské číslice a každý nižší stupeň podřazenosti je posunut o další číslice vpravo. Číslice jsou členěny jednotlivě nebo do trojčíslí a oddělují se mezerami nebo tečkami. Vyrovnávají se tak, aby číslice označující stupeň podřazenosti byly přesně pod sebou. (Pop, Flégr, Pop, 1989)

V obsahu knihy je text s čísly stran spojen tečkami. Pro tento účel se používají tečky odlévané většinou na čtverčík, někdy i na půlčtverčík. Obraz tečky je umístěn uprostřed nebo v pravém okraji účaří písma. Přechází-li text do dvou nebo i více řádek, spojuje se s číslicí až konec textu. (Pop, Flégr, Pop, 1989)

Mezi poslední literou textu a první tečkou musí být mezera nejméně ve velikosti třetiny čtverčíku, mezi poslední tečkou a nejdelší číslicí má být mezera alespoň půlčtverčík. Od klasické koncepce sazby obsahu s tečkami se v poslední době upouští. (Pop, Flégr, Pop, 1989)

Sazba číslicová – tisíce a milióny se v sazbě číslic oddělují mezerou v šířce tečky. U letopočtu se číslice neoddělují. (Pop, Flégr, Pop, 1989)

V desetinných číslech se oddělují desetinná místa desetinnou čárkou. (Pop, Flégr, Pop, 1989)

Pokud se řadí číslice do sloupců pod sebou, jednotky, desítky, stovky, stejně tak i desetinná čísla se sázejí přesně pod sebe. Ve výrazech, kde jsou spojeny číslo a písmeno, sázejí se oba prvky bez mezery. (Pop, Flégr, Pop, 1989)

I když v případě pořadové sazby nejde přímo o sazbu tabulek, bylo by vhodné, aby v budoucím stylu pro sazbu tabulek byla implementována také tvorba pořadové sazby, která by se používala stejně nebo velmi podobně jako sazba tabulek a zvolení pořadové sazby by byla pouze otázka nastavení.

3.2 Tabulky

Sazba tabulek navazuje na sazbu pořadovou. Je to sazba textu a číslic přehledně upravená linkami. Pro svou náročnost vyžaduje od sazeče všestrannou odbornost, navíc pak cit pro estetiku z hlediska celkové grafické úpravy. Tabulky se vyskytují v mnoha tiskovinách s různými obměnami, někde ve formě samostatných tiskovin jednostranně nebo oboustranně potištěných, jindy jsou doplňkem knih, brožur, časopisů i novin. Některé jsou velmi složité z hlediska zpracování, naproti tomu však existují tabulky jednoduché, lišící se od pořadové sazby pouze linkami oddělujícími sloupce číslic nebo textu od hlavičky, popř. součtu. (Pop, Flégr, Pop, 1989)

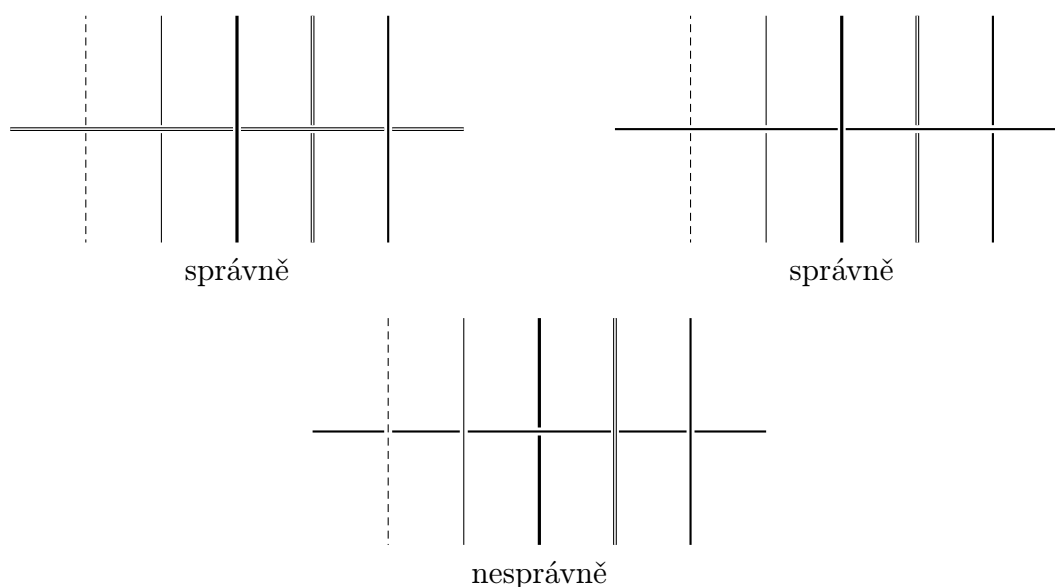
přizpůsobil rukopis typografickému měrnému systému. V podstatě žádné pravidlo neuvádí, o kolik bodů může v jednotlivých kolonkách přidávat nebo ubírat. Záleží vždy na estetickém cítění i zkušenostech pracovníka a na tom, aby se hotový otisk podobal co nejvíce originálu. (Pop, Flégr, Pop, 1989)

Jestliže sazeč vychází z tištěného vzoru, musí dodržovat přesné rozměry všech rubrik i celé tabulky. (Pop, Flégr, Pop, 1989)

U tabulek, jež se po použití zařazují do pořadníků, má zůstat nepotištěný levý okraj v šíři 20 mm. V některých případech je také nutné ponechat nad tabulkou i pod ní dostatek volného místa pro zvláštní záznamy vztahující se k tabulce. Za základ k výpočtu tabulek z rukopisné předlohy slouží hlavička tabulky. (Pop, Flégr, Pop, 1989)

Stupeň a řez písma použité v tabulkách se řídí několika faktory: samotným účelem tabulky, jejími rozměry a množstvím textu, který obsahuje. Pro použitý řez písma je rozhodující grafická úprava tiskoviny jako celku a významnou roli zde hraje tloušťka linek použitých v tabulce. Běžně používané stupně písma pro sazbu tabulek jsou nonpareille, petit a garmond. (Pop, Flégr, Pop, 1989)

V tabulkách se vedení a křížení linek řídí přesnými pravidly. **Linky obrazem tlustší přetínají linky tenčí ve vodorovném i svislém směru.** Pořadí linek podle kresby obrazu je tedy: tečkovaná, jednoduchá, tupojemná, dvojité, polotučná, tučná. Pokud se kříží linky stejného obrazu, má přednost linka vodorovná před svislou: základní linka, linka pod hlavičkou, součtová linka a patní (spodní) linka u uzavřené tabulky. V nohách tabulky u vsazeného příčnicku mají vždy přednost linky svislé před vodorovnými, protože tento způsob umožňuje snazší technickou práci s materiálem, omezuje spotřebu kratších linek a zaručuje lepší estetický vzhled tisku. (Pop, Flégr, Pop, 1989)



Obr. 2: Křížení linek. Volně překresleno z (Pop, Flégr, Pop, 1989)

Uvedená pravidla pro křížení linek je nutné v budoucím stylu respektovat. V ideálním případě by měl styl v těchto záležitostech rozhodovat sám a uživatel by tak dodržováním těchto pravidel nebyl vůbec zatěžován, neboť by se o správnost staral sám styl. Za vyhovující ale lze označit i stav, kdy uživatel má možnost tato pravidla respektovat dodatečným nastavením linek, čímž se myslí obecnější nastavení (např. jedna změna v nastavení pro všechny vodorovné linky), ale také nastavení jediné konkrétní linky. V žádném případě však problematika křížení linek nesmí zůstat nevyřešena.

3.2.1 Typy tabulek

Tabulky se dělí podle úpravy, účelu a technického zpracování. (Pop, Flégr, Pop, 1989)

Podle úpravy se tabulky dělí na pořadově sázené (bez linek a s linkami), otevřené, uzavřené a převislé. (Pop, Flégr, Pop, 1989)

Pořadově sázené tabulky se vyskytují v novinách, časopisech, brožurách. Tato sazba je vlastně přechodem mezi sazbou pořadovou a sazbou tabulkovou. Sáží se ze stejného řezu i stupně, jako je základní text, popř. z písma o jeden stupeň menšího. Čísla ve sloupcích se sázejí pod sebou podle hodnot (jednotky, desítky, stovky atd.) při zachování stejných mezer mezi sloupci. Rovnoměrnosti lze dosáhnout tak, že nejprve se vysadí nejširší řádka a současně se stanoví mezery mezi sloupci. Opakující se výrazy sázené pod sebou se nahrazují uvozovkami, sázenými rovněž pečlivě pod sebou. (Pop, Flégr, Pop, 1989)

Text v hlavičce se sází až po vysazení jednotlivých číslicových nebo textových sloupců, pokud možno na plný počet cicer, a to na šířku i na výšku hlavičky. Celková šířka tabulky musí být na plný počet cicer. (Pop, Flégr, Pop, 1989)

PRŮMĚR PROSPĚCHU ODBORNÝCH SKUPIN

Odborná skupina	Čeština	Občanská výchova	Ruština	Matematika	Fyzika	Chemie	Tělesná výchova	Branná výchova	Kreslení	Odborný výcvik
I.	2,24	2,13	1,15	2,43		1,51	2,00	2,41		1,21
II.	1,45	1,20	2,05	2,17		2,27	2,15	1,14	1,31	1,37
III.	1,98	2,75	2,99	1,72	1,50	2,57	2,19	2,49	1,33	1,71
IV.	2,24	2,57	1,39	1,11	1,61	2,84	2,11	1,63	1,70	2,51

Obr. 3: Pořadová sazba v tabulkové úpravě. Volně překresleno z (Pop, Flégr, Pop, 1989)

Tabulky otevřené nejsou uzavřeny svislými linkami po stranách, ale pouze linkou v záhlaví, výjimečně se vyskytují s příčnou uzavírací linkou v patě tabulky.

Jsou vhodné pro tisk na spadávání, tj. od levého okraje papíru k pravému. (Pop, Flégr, Pop, 1989)

Obr. 4: Tabulka otevřená. Volně překresleno z (Pop, Flégr, Pop, 1989)

Tabulky uzavřené se používají zpravidla ve všech hospodářských tiskovinách, knihách, časopisech i akcidenčních pracích. Jsou ohraničeny linkami stejné tloušťky na všech stranách. Pokud se sází rámeček z tučných linek, dává se vodorovná linka přes celou šíři tabulky a na ni navazují boční svislé linky. (Pop, Flégr, Pop, 1989)

V některých případech se sází rámeček z polotučných linek odlitých na okraj čtyřbodové až dvanáctibodové kuželky. (Pop, Flégr, Pop, 1989)

Obr. 5: Tabulka uzavřená. Volně překresleno z (Pop, Flégr, Pop, 1989)

Tabulky oboustranně převislé se uplatňovaly hlavně při tisku účetních knih, faktur apod. Záhlavní linka u těchto tabulek přečnívá na obou stranách šíři vnitřní sazby tabulky. Dnes se používají velmi omezeně, a to vzhledem k stále širšímu uplatňování výpočetní techniky, a tím i tiskopisů pro ni určených. (Pop, Flégr, Pop, 1989)

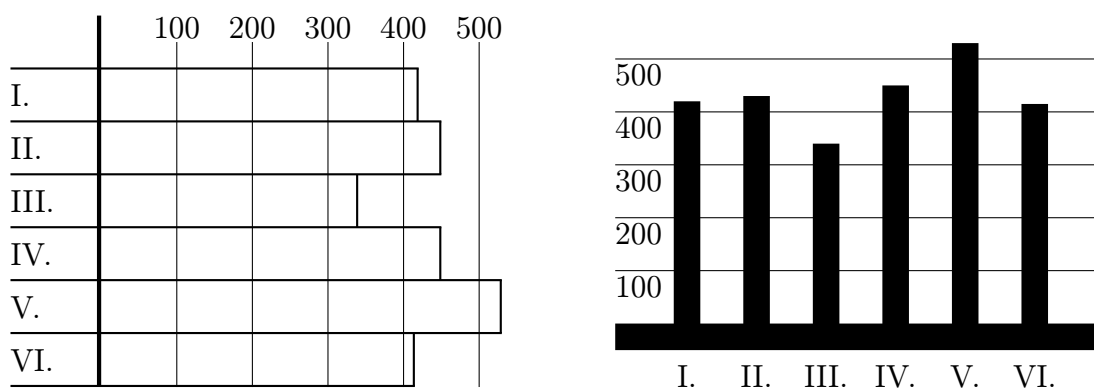
Tabulky jednostranně převislé se obvykle sázejí jako dvě protilehlé tabulky do středu převislé, čímž dvě protilehlé strany rozdělené hřbetní mezerou tvoří jeden celek. Také tyto tabulky se používají jen zřídka. (Pop, Flégr, Pop, 1989)

Budoucí styl by měl umožňovat nastavení v takové míře, aby v něm byla snadno řešitelná tvorba více typů tabulek.

3.3 Diagramy

Budoucí styl nebude přímo určen pro sazbu diagramů. Pokud by však ale zvolená implementace umožňovala tvorbu alespoň některého typu diagramu podobným způsobem jako tvorbu tabulek nebo pořadové sazby, není důvod uživateli nenabídnout toto další využití tohoto stylu.

Diagramy – jejich účelem je získat rychlou a názornou orientaci při sledování určitého vývoje. Čísla jsou v diagramu přeměněna na grafické znázornění, přičemž v plošném uspořádání umožňují jasný přehled. Nejčastěji užívané jsou diagramy sloupcové, sázené na výšku i šířku. Při jejich sazbě je nutný přesný rozpočet metrických měr na typografickou míru i určení přesného základu, z něhož se vychází. U plošných diagramů se velikost hodnot znázorňuje velikostí plochy. (Pop, Flégr, Pop, 1989)



Obr. 6: Sloupcový diagram na šířku. Sloupcový diagram na výšku. Oba diagramy volně překresleny z (Pop, Flégr, Pop, 1989)

3.4 Prostředí pro sazbu tabulek v typografickém systému \TeX / \LaTeX

Popisem prostředí a balíků pro sazbu tabulek se mj. podrobně zabývá (Talandová, 2006). Zde jsou uvedeny pouze některé.

3.4.1 Prostředí `tabbing`

Pro pořadovou sazbu je v systému L^AT_EX k dispozici prostředí `tabbing`. (Rybička, 2003)

Prostředí `tabbing` umožňuje rozmístit části textu ve vodorovném směru systémem podobným systému tabulačních zářezů na psacím stroji. Jsou zde k dispozici příkazy pro nastavení tabulační zářezky, pro přechod sazby k následující nastavené zářezce, pro posun levého okraje k následující zářezce a podobně. Zářezky tabulátoru si pro přehlednost označíme čísly 0, 1, 2 atd. Říkáme, že zářezka číslo i je *nastavena*, když je jí přiřazena horizontální pozice na stránce. Zářezka číslo 0 je vždy nastavena na efektivní levý okraj (to je levý okraj platný na začátku prostředí). Je-li nastavena zářezka i , pak jsou také nastaveny všechny zářezky od 0 do $i - 1$. Zářezka číslo $i - 1$ je umístěna vždy vlevo od zářezky i . (Rybička, 2003)

Uvnitř prostředí `tabbing` se může objevit konec stránky, délka prostředí tedy není omezena. Tento způsob zarovnání textu se často používá mimo jiné k formátování zápisu strukturovaných programů (například v jazyce Pascal). (Rybička, 2003)

Účinek tabulačních příkazů lze popsat hodnotami dvou veličin zvaných *další zářezka tabulátoru* (označíme D) a *levá okrajová zářezka* (označíme L). Na začátku je hodnota D rovna 1, hodnota L je 0 a je nastavena pouze zářezka číslo 0. Hodnota D je zvyšována příkazy `\>` a `\=` a znovu nastavena na počátek příkazy `\|` a `\kill`. (Rybička, 2003)

Tab. 1: Příklad pořadové sazby vytvořené v prostředí `tabbing`

Seznam účastníků poháru žáků v běhu

Číslo	Jméno	Trať	Rok narození	Město
1	Bárta Miroslav	100 m	2001	Břeclav
2	Bylný Milan	100 m, 200 m	2000	Brno
3	Kadlec Jaromír	200 m	2000	Vyškov
4	Krusta Ondřej	200 m	2000	Brno
5	Novák Jan	100 m	1999	Brno
6	Novák Tomáš	100 m	1999	Kuřim
7	Pavelka Petr	100 m, 200 m	2001	Brno
8	Sup Václav	100 m, 200 m	2000	Tišnov
9	Štrbák David	200 m	2001	Brno
10	Žák Michal	100 m, 200 m	2000	Blansko

Tato pořadová sazba byla získána následujícím zápisem:

```
\begin{tabbing}
\textbf{Číslo} \= Kadlec Jaromír \= 100 m, 200 m \=
\textbf{Rok narození} \= Břeclav\kill\
```

```

\textbf{Seznam účastníků poháru žáků v běhu}\\
\textbf{Číslo} \> \textbf{Jméno} \> \textbf{Trať} \>
\textbf{Rok narození} \> \textbf{Město}\\
1 \> Bárta Miroslav \> 100 m \> 2001 \> Břeclav\\
2 \> Bylný Milan \> 100 m, 200 m \> 2000 \> Brno\\
3 \> Kadlec Jaromír \> 200 m \> 2000 \> Vyškov\\
4 \> Krusta Ondřej \> 200 m \> 2000 \> Brno\\
5 \> Novák Jan \> 100 m \> 1999 \> Brno\\
6 \> Novák Tomáš \> 100 m \> 1999 \> Kuřim\\
7 \> Pavelka Petr \> 100 m, 200 m \> 2001 \> Brno\\
8 \> Sup Václav \> 100 m, 200 m \> 2000 \> Tišnov\\
9 \> Štrbák David \> 200 m \> 2001 \> Brno\\
10 \> Žák Michal \> 100 m, 200 m \> 2000 \> Blansko\\
\end{tabbing}

```

3.4.2 Prostředí tabular

Pro sazbu tabulek je v systému L^AT_EX k dispozici prostředí `tabular`. Umožňuje snadné vytváření tabulek včetně čar, přičemž je možné v každém sloupci používat jiný způsob svislého zarovnání položek. Celá tabulka se chová jako jeden box (nelze tedy uprostřed tabulky přejít na novou stránku), který se k předchozímu a následujícímu textu připojuje středem, horním, nebo spodním okrajem. (Rybička, 2003)

Prostředí má dvě varianty – základní variantu a variantu s hvězdičkou. V základní variantě šířka tabulky odpovídá obsahu jednotlivých sloupců, zatímco varianta s hvězdičkou umožňuje vytvořit tabulku, jejíž celková šířka je zadána dalším povinným parametrem. (Rybička, 2003)

Definice způsobu zarovnání textů v jednotlivých sloupcích, počet sloupců a svislých čar mezi nimi se provádí v parametru *sloupce*. Tento parametr se skládá z písmen určujících zarovnání sloupců, znaku svislé čáry (pipe) pro umístění svislých čar v tabulce a výrazů začínajících znakem @ pro definici jiné mezisloupcové výplně. (Rybička, 2003)

Pro definici několika stejných sloupců nebo stejných skupin sloupců lze s výhodou použít násobnou definici s opakovacím ve tvaru `*{kolik}{co}`. (Rybička, 2003)

Jednotlivé položky tabulky jsou v řádku od sebe odděleny znakem `&`, na konci každého řádku je příkaz `\\`. Pokud je obsahem posledního sloupce text, v němž mohou být příkazy `\\` pro jeho zlom, pak konec řádku tabulky lze nařídit příkazem `\tabularnewline[míra]`, kde volitelný parametr *míra* udává stejně jako u příkazu `\\` svislou vzdálenost od následujícího řádku. Počet položek v každém řádku musí přesně souhlasit s počtem zarovnávacích písmen v hlavičce tabulky. Výjimkou je případ, kdy v posloupnosti prázdných políček tabulky na konci řádku vynecháme přebytečné oddělovače `&`. Pokud však zde mají být svislé čáry, budou vynechány. (Rybička, 2003)

Vodorovné čáry se nařizují na konci řádku příkazem `\hline`. Pokud čára nemá vést přes celou šířku tabulky, použijeme příkaz `\cline{x-y}`, kde x je pořadové číslo prvního sloupce a y pořadové číslo posledního sloupce, přes který má vodorovná čára jít. (Rybička, 2003)

Mezi sloupci se automaticky vytváří mezisloupcová mezera, jejíž šířku lze ovládat ve specifikaci sloupců výrazem začínajícím znakem `@`. Tento výraz má jeden povinný parametr, který určuje, co bude vloženo mezi příslušné dva sloupce. Chceme-li například, aby mezi dvěma sloupci byl místo svislé čáry znak „=“, zapíšeme do hlavičky tabulky výraz `@{=}`. Uvedením `@`-výrazu se veškeré mezisloupcové mezery ruší a nahrazují se obsahem parametru. Mezi sloupci s uvedeným výrazem tedy bude jen znak rovnítko a žádná jiná mezera. U prostředí `tabular*` je možné tímto výrazem nastavit i potřebnou mezeru mezi sloupci, aby se šířka tabulky doplnila na požadovaný rozměr. Zapíšeme-li `@{\extracolsep{\fill}}` mezi první dva sloupce tabulky, budou všechny mezisloupcové mezery zvětšeny o potřebnou míru. (Rybička, 2003)

Chceme-li jen zrušit mezisloupcovou mezeru, zapíšeme `@{}`. (Rybička, 2003)

Předdefinovaná velikost vzdálenosti mezi vnitřkem sloupce a jeho okrajem je 6 pt. Tuto míru představuje délkový registr `\tabcolsep`. Chceme-li v celé tabulce tuto míru změnit například na 3 pt, stačí před tabulkou uvést příkaz `\tabcolsep=3pt`. (Rybička, 2003)

Tab. 2: Příklad tabulky vytvořené v prostředí `tabular`

Seznam účastníků poháru žáků v běhu				
Číslo	Jméno	Trať	Rok narození	Město
1	Bárta Miroslav	100 m	2001	Břeclav
2	Bylný Milan	100 m, 200 m	2000	Brno
3	Kadlec Jaromír	200 m	2000	Vyškov
4	Krusta Ondřej	200 m	2000	Brno
5	Novák Jan	100 m	1999	Brno
6	Novák Tomáš	100 m	1999	Kuřim
7	Pavelka Petr	100 m, 200 m	2001	Brno
8	Sup Václav	100 m, 200 m	2000	Tišnov
9	Štrbák David	200 m	2001	Brno
10	Žák Michal	100 m, 200 m	2000	Blansko

Tato tabulka byla získána následujícím zápisem:

```
\begin{tabular}{|l|l|l|l|l|}\hline
\multicolumn{5}{|l|}{\textbf{Seznam účastníků poháru žáků v běhu}}
\\ \hline
\textbf{Číslo} & \textbf{Jméno} & \textbf{Trať} &
\textbf{Rok narození} & \textbf{Město} \\ \hline
1 & Bárta Miroslav & 100 m & 2001 & Břeclav \\ \hline
2 & Bylný Milan & 100 m, 200 m & 2000 & Brno \\ \hline\end{tabular}
```

```

3 & Kadlec Jaromír & 200 m & 2000 & Vyškov\\hline
4 & Krusta Ondřej & 200 m & 2000 & Brno\\hline
5 & Novák Jan & 100 m & 1999 & Brno\\hline
6 & Novák Tomáš & 100 m & 1999 & Kuřim\\hline
7 & Pavelka Petr & 100 m, 200 m & 2001 & Brno\\hline
8 & Sup Václav & 100 m, 200 m & 2000 & Tišnov\\hline
9 & Štrbák David & 200 m & 2001 & Brno\\hline
10 & Žák Michal & 100 m, 200 m & 2000 & Blansko\\hline
\end{tabular}

```

3.4.3 Balík colortbl

Často se pro zvýraznění určitých částí tabulky vybarvují buňky podkladovými barvami nebo rastry. Z typografického hlediska se nejedná o klasický nástroj, neboť v horké sazbě byl velmi pracný a standardními prostředky těžko řešitelný. Přesto jistě patří k moderním způsobům úpravy tabulek. Je však potřebné dodržovat obecnou zásadu, že méně je více. Není účelné, když plocha tabulky hraje všemi barvami. Přehlednost a funkčnost je tím spíše snížena. Jako doplněk, například zvýraznění záhlaví nebo důležitých buněk, můžeme s výhodou podbarvení použít. Tuto funkci řeší balík `colortbl` přítomný v běžných distribucích systému \TeX . (Rybička, 2003)

3.5 Přístup k tvorbě tabulek v jazyce HTML

K tvorbě dokumentů pro WWW bylo nutné vytvořit vývojový nástroj, který by umožnil formátovat text, vkládat do textu odkazy atd. Proto se v roce 1991 objevila první verze jazyka HTML (*HyperText Markup Language*). Tento jazyk slouží k popisu toho, co má být na stránce umístěno a jak má stránka vypadat. (Písek, 2006)

HTML dokument se od obyčejného textového dokumentu liší tím, že kromě vlastního obsahu stránky obsahuje navíc informace o vzhledu a formátování stránky, které jsou určené pro prohlížeč. (Písek, 2006)

Těžko si lze představit stránku prezentující nějakou společnost bez tabulek. (Písek, 2006)

Tabulku můžete vložit do dokumentu pomocí značek `<TABLE>` a `</TABLE>`. Každý řádek tabulky je označen dvojicí značek `<TR>` a `</TR>` a každá buňka v řádku je obklopena dvojicí značek `<TD>` a `</TD>`. Pokud označíte buňky značkami `<TH>` a `</TH>`, budou tvořit záhlaví tabulky. (Písek, 2006)

Práce s tabulkami v jazyce HTML je podrobně popsána v (Písek, 2006).

Příklad zápisu tabulky uvedené u popisu prostředí `tabular` v jazyce HTML:

```

<table border=1>
<tr> <td colspan=5><b>Seznam účastníků poháru žáků v běhu</b>
</td></tr>

```

```

<tr> <td><b>Číslo</b></td> <td><b>Jméno</b></td> <td><b>Trať</b></td>
<td><b>Rok narození</b></td> <td><b>Město</b></td> </tr>
<tr> <td>1</td> <td>Bárta Miroslav</td> <td>100 m</td>
<td>2001</td> <td>Břeclav</td> </tr>
<tr> <td>2</td> <td>Bylný Milan</td> <td>100 m, 200 m</td>
<td>2000</td> <td>Brno</td> </tr>
<tr> <td>3</td> <td>Kadlec Jaromír</td> <td>200 m</td>
<td>2000</td> <td>Vyškov</td> </tr>
<tr> <td>4</td> <td>Krusta Ondřej</td> <td>200 m</td>
<td>2000</td> <td>Brno</td> </tr>
<tr> <td>5</td> <td>Novák Jan</td> <td>100 m</td>
<td>1999</td> <td>Brno</td> </tr>
<tr> <td>6</td> <td>Novák Tomáš</td> <td>100 m</td>
<td>1999</td> <td>Kuřim</td> </tr>
<tr> <td>7</td> <td>Pavelka Petr</td> <td>100 m, 200 m</td>
<td>2001</td> <td>Brno</td> </tr>
<tr> <td>8</td> <td>Sup Václav</td> <td>100 m, 200 m</td>
<td>2000</td> <td>Tišnov</td> </tr>
<tr> <td>9</td> <td>Štrbák David</td> <td>200 m</td>
<td>2001</td> <td>Brno</td> </tr>
<tr> <td>10</td> <td>Žák Michal</td> <td>100 m, 200 m</td>
<td>2000</td> <td>Blansko</td> </tr>
</table>

```

3.6 Přístup k tvorbě tabulek v textovém procesoru Microsoft Word

V kancelářské praxi budete často nuceni k tomu, abyste text řadili do sloupců. Nejprve vždy zvažte, zda nebude výhodnější použít pro takový dokument vhodnější program, kterým je *MS Excel*. Obsahuje-li dokument velké množství číselných hodnot, mezi nimiž existují matematické vztahy, je na místě *Word* opustit a pracovat s tabulkovým procesorem. (Pecinovský, 2007)

Jakmile budoucí sloupce obsahují jednoduchá čísla, nejsou tu žádné nebo minimální matematické vztahy, nebo je úsek řazený do sloupců krátký, vystačíte si s *Wordem*. (Pecinovský, 2007)

Při použití tabulátorů pracujete na normální pracovní ploše, jen část textu vždy odsunete směrem doprava pomocí tabulačních zářezek. Při práci s tabulkou rozdělíte pracovní plochu mřížkou na jednotlivé buňky, seřazené do řádků a sloupců. Navíc tu máte možnost mřížku zobrazit, takže vytvoříte skutečnou tabulku. (Pecinovský, 2007)

Doslova barbarstvím je formátování textu do sloupců s použitím mezerníku, jak bohužel ještě často můžeme vidět. Takový text se nikdy nepodaří zarovnat do

sloupců přesně, je to velmi pracné, a navíc je celá struktura náchylná ke zhroucení.
(Pecinovský, 2007)

Práce s tabulátory a tabulkami je podrobně popsána v (Pecinovský, 2007).

4 Analýza existujících dokumentů obsahujících tabulky a definice požadavků

Při tvorbě této kapitoly byly používány (Janovský, 2012a), (Janovský, 2012b) a (Janovský, 2012c). Tyto zdroje popisují tvorbu a formátování tabulek v jazyce HTML.

4.1 Analýza existujících dokumentů

Analýzou množství existujících dokumentů obsahujících tabulky je třeba zjistit, v jaké podobě se v nich tabulky vyskytují. K tomu bude nutné nejdříve sestavit výčet vlastností a formátování tabulek, jejichž užívání bude v tabulkách sledováno. Poté proběhne vyhodnocení analýzy. Teprve na základě tohoto vyhodnocení bude možné definovat požadavky na styl. Důraz bude kladen na stanovení co nejkonkrétnějších požadavků, což by bylo bez zmíněné analýzy problematické.

Samozřejmě existují požadavky, které jsou zcela zřejmé a analýza k jejich zjištění není nutná. Budou uvedeny později společně s požadavky, které vyplynou z analýzy existujících dokumentů.

4.1.1 Sledované vlastnosti a formátování

- formátování tabulky jako celku
 - užívání titulku tabulky
 - užívání poznámek pod tabulkou
 - užívané velikosti písma v tabulce
 - užívání tabulky s přesahem na více stran
- formátování řádků a sloupců
 - užívání formátování společného pro celý sloupec nebo řádek (příp. pro více po sobě následujících sloupců nebo řádků)
- formátování buněk
 - horizontální zarovnání obsahu buňky
 - vertikální zarovnání obsahu buňky
 - slučování buněk
 - * přes více sloupců
 - * přes více řádků
 - barevné pozadí buňky
 - rámeček buňky
 - přeškrtnutí buňky
- formátování linek
 - použité šířky linek
 - barvy linek
 - styly linek

4.1.2 Výsledky analýzy

Analýza byla provedena u třiceti různých dokumentů obsahujících tabulky. Většina z nich je ve formátu pdf, některé jsou ve formátu doc.

K analýze užívaných formátování tabulek byly využity dokumenty, jejichž seznam je uveden v příloze A.

Na výše uvedené sledované vlastnosti se je třeba ve výsledcích zaměřit jednotlivě.

Titulek tabulky

Titulek tabulky se v mnoha dokumentech používá. Jen velmi vzácně bylo ale součástí titulku také číslo tabulky. Pokud byl titulek u tabulky použit, byl vždy umístěn nad tabulkou horizontálně zarovnaný vlevo nebo na střed.

Poznámky pod tabulkou

Poznámky pod tabulkou jsou v některých analyzovaných dokumentech používány. Často jsou sázeny nižším stupněm písma než které je použito v tabulce.

Velikost písma v tabulce

Velmi časté je použití jednoho nebo dvou různých stupňů písma v celé tabulce. Ovšem i v případě, kdy jsou použity dva stupně písma, převážná většina textu v tabulce je vysázena jedním stupněm písma.

Tabulky s přesahem na více stran

Většina analyzovaných dokumentů obsahuje tabulky začínající i končící na stejné straně. Vícestránkové tabulky se ale přesto objevují relativně často.

Formátování společné pro celý sloupec nebo řádek (příp. pro více po sobě následujících sloupců nebo řádků)

Nejběžnější je užívání horizontálního zarovnání textu společné pro celý nebo téměř celý sloupec. Velmi často užívaný způsob formátování celých sloupců i řádků (nebo téměř celých) je stejná barva pozadí buňky.

Slučování buněk přes více sloupců a řádků

Slučování buněk přes více sloupců je velmi časté. Slučování buněk přes více řádků je již používáno méně.

Barevné pozadí buňky

Barevné pozadí buněk v tabulkách v analyzovaných dokumentech je používáno často. Užíváno je mnoho barev v mnoha odstínech.

Rámeček buňky

Užívání rámečků buněk je v analyzovaných dokumentech velmi vzácné. V některých analyzovaných dokumentech ale použity jsou.

Přeškrtnutí buňky

Použití přeškrtnutí buňky nebylo objeveno v žádném z analyzovaných dokumentů.

Formátování linek

V analyzovaných dokumentech se téměř nikdy nepoužívají v tabulkách více než dvě různé šířky linek. Barva linek je většinou černá, ale práce s barvami je nezbytná z důvodu častého užívání barevného pozadí buňky. Jen v několika případech se používá jiný styl linky než plný.

4.2 Definice požadavků

Jak již bylo zmíněno, některé požadavky jsou zcela zřejmé a není nutné k jejich zjištění vytvářet analýzu. Další požadavky byly definovány na základě analýzy a jiné již byly v této práci zmíněny dříve. Definice požadavků na styl je tedy následující:

- tvorba různých tabulek
 - možnost tvorby pořadové sazby
 - možnost tvorby různých typů tabulek
- formátování tabulky jako celku
 - možnost nastavení vertikální mezery nad tabulkou a pod ní
 - možnost uvedení titulku tabulky vertikálně umístěným nad tabulkou, včetně možnosti číslování tabulek
 - možnost tvorby poznámek pod tabulkou
 - možnost přednastavení stupně písma použitého v tabulce, titulku a poznámek pod tabulkou
- formátování sloupců a řádků
 - nastavení šířky sloupce včetně možnosti automatického určení šířky buněk podle nejširšího obsahu buňky v daném sloupci
 - možnost nastavení šířky mezisloupcové mezery
 - nastavení výšky řádku včetně možnosti automatického určení výšky buněk podle nejvyššího obsahu buňky v daném řádku
 - možnost nastavení šířky meziřádkové mezery
 - možnost nastavení výchozího horizontálního zarovnání buněk ve sloupci
- formátování buněk
 - volba horizontálního zarovnání obsahu buňky
 - volba vertikálního zarovnání obsahu buňky
 - nastavení vzdálenosti obsahu buňky od jejích okrajů
 - možnost slučování buněk přes více sloupců
 - možnost nastavit barvu pozadí buňky
- formátování linek
 - možnost nastavení šířky linek včetně možnosti si dvě různé šířky linek přednastavit
 - možnost nastavit překrývání křížících se linek

Tyto požadavky mají nejvyšší prioritu a na jejich zahrnutí bude v implementaci stylu kladen velký důraz. Následující požadavky mají prioritu nižší, ale jejich zahrnutí do implementace přispěje k vyšší komplexnosti a použitelnosti stylu:

- možnost využití stylu pro sazbu některých diagramů
- možnost vytvářet tabulky s přesahem na více stran
- možnost nastavení co největšího množství parametrů, které budou použity pro výchozí formátování konkrétního sloupce nebo řádku
- možnost slučování buněk přes více řádků
- možnost sazby rámečku buňky
- možnost přeškrtnutí buňky
- možnost nastavení barvy linek
- možnost nastavení jiných stylů linek

5 Výběr implementačních nástrojů

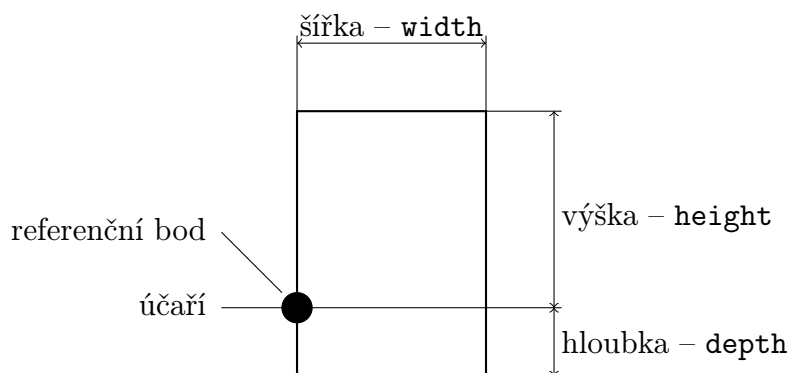
K tvorbě stylu budou potřeba nástroje pro dva základní úkony:

- nástroj k rozmísťování textů (titulku tabulky, obsahu buněk a poznámek pod tabulkou),
- nástroj pro kresbu všech grafických prvků tabulky (všech linek, barevného pozadí buněk, rámečku buněk a přeškrtnutí buněk). Je požadována také podpora práce s různými barvami.

5.1 Boxy

Pojem *box*¹ představuje základní prvek sazby v systému $\text{T}_{\text{E}}\text{X}$. Je elektronickou obdobou čela (horní plochy) kovové litery, na níž je odlit reliéf určitého znaku. Jako je principem kovové sazby „sázení“ jednotlivých liter vedle sebe, principem elektronické sazby je (v podání systému $\text{T}_{\text{E}}\text{X}$) skládání boxů těsně vedle sebe, a to tak, aby jejich referenční body ležely v jediné lince – na účaří. (Rybička, 2003)

Boxy se do sebe mohou přirozeným způsobem vkládat, například tabulka (vnější box) obsahuje boxy s jednotlivými řádky a v těchto řádcích jsou seskládány boxy s jednotlivými znaky. (Rybička, 2003)



Obr. 7: Box a jeho rozměry. Volně překresleno z (Rybička, 2003)

Uvnitř boxu nemůže nastat konec stránky, uvnitř některých boxů ani konec řádku. Kromě boxů, které $\text{T}_{\text{E}}\text{X}$, popřípadě $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ vytváří automaticky, si může uživatel určitý materiál jako box označit podle potřeby. Práce s boxy je důležitým nástrojem při vytváření složitějších kompozic textu na stránce. $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ rozeznává tři druhy boxů: tzv. LR boxy, parboxy a rule boxy. (Rybička, 2003)

¹Jazyková čistota zde patrně utrpí, avšak překládat tento pojem do češtiny nebudeme – náhradní výrazy, jako například rámeček nebo obdélník, nejsou tak užívané a jasné. (Rybička, 2003)

5.1.1 LR boxy

LR mód (Left to Right) je způsob skládání textu do řádku zleva doprava. Je to L^AT_EXový pojem pro tzv. omezený horizontální mód (restricted horizontal mode), v němž nemůže nastat konec řádku. Tento box je tedy určen pro text, který sestává z jednoho řádku. (Rybička, 2003)

5.1.2 Parboxy

Parbox je box, jehož obsah je zpracováván v horizontálním režimu jako běžné odstavce (paragraphs). Například prostředí `figure` nebo `table` produkují parboxy. Při lámání textu do řádků je potřeba vědět, jaká má být délka řádku. Proto je u parboxu vždy povinný parametr s touto specifikací. (Rybička, 2003)

Omezení rozsahu textu buňky na jeden řádek je příliš velké a ve víceřádkových buňkách by se LR boxy musely používat opakovaně, což by bylo příliš komplikované. Bude tedy nutné při implementaci používat pro rozmísťování textu parboxy.

5.1.3 Rule boxy

Rule box je černý obdélník. Je vytvářen příkazem `\rule`, jehož argumenty definují rozměry obdélníka. (Rybička, 2003)

Čára o šířce 0 pt je nazývána *strut*. Tato čára je neviditelná, ale její rozměry T_EX započítává při sazbě okolního textu. Toho lze využít pro vymezení většího prostoru kolem určitého místa, například v tabulkách, rámečcích a podobně. (Rybička, 2003)

5.2 Volba nástroje pro sazbu grafických prvků

5.2.1 Prostředí picture


Prostředí je určeno pro kreslení vektorových obrázků složených z přímých čar, šipek, kružnic, oválů a libovolných textů. Uvnitř tohoto prostředí se mohou objevit pouze příkazy pro vložení obrazového objektu (`\put` a `\multiput`) a příkazy pro nastavení velikosti a typu písma nebo délek. (Rybička, 2003)

Celé prostředí se chová jako box, tj. stejně jako jediné písmeno. Všechny obrazové objekty se umísťují v pravoúhlém souřadnicovém systému, jehož počátek je v levém dolním rohu (pracovní plocha se nachází v prvním kvadrantu). Počátek souřadnic lze lineárně posunout do libovolného jiného bodu. Všechny hodnoty souřadnic a jiných rozměrů uvnitř prostředí `picture` jsou chápány jako násobky jednotkové délky. Ta je uschována v délkovém registru `\unitlength`. a je třeba ji ještě před začátkem prostředí nastavit. Často se používá milimetrové měřítko nastavením na 1 mm. Pokud není základní jednotka nastavena, je `\unitlength=1pt`. (Rybička, 2003)

5.2.2 Balík color

Tento balík nenabízí přímo práci s obrázky, ale práci s barvami u všech objektů – textových i grafických. Barvy lze vybírat podle toho, jaký barevný model zvolíme. K dispozici je standardně barevný model **rgb**. Jeho barva je definována trojicí čísel v rozsahu od nuly do jedné, které specifikují jas červené, zelené a modré složky dané barvy. Hodnota 1 znamená maximální jas. Černá barva je například definována hodnotami 0, 0, 0. Druhý model – **gray** – umožňuje pracovat s odstíny šedi. Definice barvy tedy obsahuje jediné číslo od nuly do jedné. Třetí standardní model je **cmk** se čtyřmi složkami cyan, magenta, yellow, black. Následující barvy jsou předdefinovány: **black**, **white**, **red**, **green**, **blue**, **yellow**, **cyan** a **magenta**. (Rybička, 2003)

5.2.3 Balíky PGF a TikZ

Balík **pgf**, kde „**pgf**“ znamená „přenosný grafický formát“ (nebo „pěkný“, „dobrý“, „funkční“, chcete-li), je balík pro tvorbu grafiky způsobem „do řádků“. Nabízí množství **T_EX**ových příkazů, které vykreslují grafiku. Např. kód `\tikz \draw (0pt, 0pt) -- (20pt, 6pt);` poskytuje čáru — a kód `\tikz \fill[orange] (1ex, 1ex) circle (1ex);` poskytuje . (Tantau aj., 2010)

V jistém smyslu, když používáte **pgf**, „programujete“ svou grafiku zrovna jako „programujete“ svůj dokument, když používáte **T_EX**. Dostanete všechny výhody „přístupu **T_EX**u k sazbě“ pro svou grafiku: rychlou tvorbu jednoduché grafiky, přesné pozicování, použití maker, často vyšší typografii. Také zdědíte všechny nevýhody: strmě učenou křivku, žádný wysiwyg, malá změna vyžaduje dlouhou dobu znovukompilace a kód skutečně „neukazuje“ jak budou věci vypadat. (Tantau aj., 2010)

K implementaci budou použity právě balíky **PGF** a **TikZ**.

6 Implementace

Během implementace byly používány zdroje (Carlisle, 2001), (Olšák, 2001), (Rybicka, 2003) a (Tantau aj., 2010).

6.1 Soubory stylu

Umístění celého stylu do jediného souboru není vhodné řešení, protože takový soubor by byl velmi rozsáhlý a bylo by obtížné v něm cokoli vyhledat.

Proto bylo zvoleno rozčlenění stylu do tří souborů. Jejich názvy a obsah jsou následující:

- Soubor `tableenvdefinitions.sty` obsahuje definice
 - základních příkazů, které se používají ve složitějších příkazech pouze jako nositelé hodnot,
 - příkazů umožňujících provádět více souvisejících nastavení na stejnou hodnotu najednou.
- Soubor `tableenvdefaults.sty` obsahuje nastavení výchozích hodnot.
- Soubor `tableenvironment.sty` je hlavním souborem stylu. Obsahuje definici
 - samotného prostředí,
 - vrstev,
 - délkových registrů,
 - čítačů (u některých také nastavení výchozí hodnoty),
 - všech složitějších příkazů.

Pokud uživatel nechce měnit nebo doplňovat funkce stylu, nemá důvod do souborů `tableenvironment.sty` (případné výjimky jsou uvedeny dále v této kapitole) a `tableenvdefinitions.sty` zasahovat. Naopak změny výchozích hodnot v souboru `tableenvdefaults.sty` jsou velmi elegantní možností, jak si tabulky přizpůsobit, pokud uživateli výchozí nastavení nevyhovují.

Všechny názvy definovaných prvků jsou založeny na slovech v angličtině (stejně jako samotné názvy souborů). Komentáře, které všechny soubory obsahují, jsou psány v češtině.

6.2 Základy stylu

Práci se stylem je možné rozčlenit do několika částí:

- Nastavení požadovaného výchozího nastavení
- Definice sloupců, mezer, mezirádkových mezer a příp. uvedení titulku tabulky
- Definice buněk, příp. definice poznámek pod tabulkou

6.2.1 Příkazy definující parametry

Ve stylu je definováno množství parametrů ovlivňujících sazbu různých prvků tabulky. Je třeba uvést, jak se s těmito parametry zachází. Při nahlédnutí do souboru `tableenvdefaults.sty` lze vidět, že je tu mnoho dvojic příkazů ovlivňujících stejný parametr. Příkazy v této dvojici se od sebe liší pouze zkratkou `def` na začátku jejich názvů. Tvar definice takovéto dvojice je tedy následující:

```
\def\prikazA#1{\gdef\@prikazA{#1}}
\def\defprikazA#1{\gdef\@defprikazA{#1}}
```

To si žádá bližší vysvětlení. V případě takovéto dvojice je první příkaz z této dvojice (tedy bez zkratky `def` na začátku názvu) využíván jako lokální parametr. Vztahuje se tedy pouze k jednomu konkrétnímu objektu. Používá se ve volitelném parametru příslušného příkazu, který slouží k sazbě prvku tabulky a tento parametr používá. Jeho osamocené použití sice nezpůsobí chybu při překladu, ale takovéto použití nemá žádný účinek.

Druhý příkaz této dvojice (tedy se zkratkou `def` na začátku názvu) je v tom případě výchozí hodnota parametru. Používá se volně, ne tedy v parametrech příkazů. Pokud není použit parametr lokální, použije se hodnota právě tohoto parametru. Pokud uživatel tento příkaz sám nepoužije, platí jeho hodnota zapsaná v souboru `tableenvdefaults.sty`. Pokud jej uživatel použije, hodnota jeho parametru platí pro všechny příslušné prvky tabulky od místa použití příkazu dále, dokud není jeho hodnota změněna s výjimkou případu, kdy je nastavení změněno lokálním parametrem.

Používání obou parametrů pro platnost konkrétnějších rozsahů je ještě rozvedeno níže.

Výsledná hodnota některého parametru (pro který existuje zmíněná dvojice příkazů) lze shrnout následujícím prioritním pořadí:

1. hodnota lokálního parametru
2. nejbližší předchozí uživatelem definovaná výchozí hodnota parametru
3. výchozí hodnota parametru definována v souboru `tableenvdefaults.sty`

Často se může stát, že uživatel bude chtít nastavit více souvisejících parametrů na stejnou hodnotu. Pro určité skupiny takových příkazů je definován další příkaz, který tuto funkci zajišťuje. Jeho použití s požadovaným parametrem tedy nahrazuje použití dvou nebo více příkazů nastavujících hodnoty po jedné. Takový zápis je kratší, přehlednější a při další změně příslušných hodnot stačí změnit hodnotu opět jediného parametru. Tyto příkazy také existují ve dvojici pro výchozí i lokální nastavení, která již byla zmíněna výše. Obecná definice takového příkazu je následující:

Pokud jsou příkazy `\prikazA` a `\prikazB` definovány takto:

```
\def\prikazA#1{\gdef\@prikazA{#1}}
\def\prikazB#1{\gdef\@prikazB{#1}}
```

Pak je možné definovat příkaz `\prikazAB` nastavující hodnoty obou příkazů najednou:

```
\def\prikazAB#1{\prikazA{#1}\prikazB{#1}}
```

Některé příkazy sloužící jako nositel hodnoty nejsou definovány ve dvojici. Některé z těchto příkazů se používají pro výchozí nebo lokální nastavení, ale druhá verze tohoto příkazu z nějakého důvodu definována není. Pokud příkaz slouží pro nastavení lokální hodnoty, může to být tím, že výchozí hodnota je získávána jinak a pokud příkaz slouží pro nastavení výchozí hodnoty, může to být tím, že se u všech prvků daného druhu očekává stejné nastavení nebo tím, že se tento prvek v tabulce použije vždy jen jednou.

Část zmíněných hodnotových příkazů slouží k zapnutí nebo vypnutí některé možnosti sazby. V takovém případě se zde používá pravidlo, že nastavení této hodnoty na 0 danou možnost vypíná (výjimečně toto nastavení znamená výchozí hodnotu). V některých případech se jedná o parametr, jehož nenulová hodnota zároveň nese hodnotu, která se při sazbě použije. Pokud daný parametr slouží jen pro zapnutí nebo vypnutí určité možnosti sazby, používá se pro zapnutí hodnota 1, přestože se zapnutí provede při zadání kterékoliv jiné hodnoty než 0. Pokud je nenulová hodnota parametru považována za délku, tak se její velikost v případě, že je hodnota nižší než 1, zapisuje do parametru příkazu pro zajištění správného vyhodnocení bez počáteční nuly, tedy např. místo 0.5 se zapíše jen .5.

6.2.2 Změna výchozího nastavení pro všechny tabulky v dokumentu

Již byla zmíněna možnost provést změnu výchozího nastavení přímo v souboru `tableenvdefaults.sty`. Tato možnost je jistě správnou volbou v případě, že uživatel chce toto nastavení používat opakovaně v různých dokumentech. Pokud ale uživatel chce tento styl použít při sazbě dokumentu, jehož tabulky mají vypadat nějakým způsobem jinak než jak si uživatel nastavil v souboru `tableenvdefaults.sty`, provádět změny v tomto souboru kvůli sazbě jednoho dokumentu již být správnou volbou nemusí. V takovém případě uživatel možná ocení možnost měnit nastavení přímo v daném dokumentu. To lze zajistit použitím libovolné podmnožiny příkazů nastavujících výchozí hodnoty (použitými v souboru `tableenvdefaults.sty`) přímo ve vlastním souboru. Hodnoty v parametrech takto použitých příkazů mají vyšší prioritu než hodnoty ve stejnojmenných příkazech v souboru `tableenvdefaults.sty` (ty v tom případě již nadále neplatí).

Tyto příkazy mohou být umístěny na téměř libovolném místě. Důležité však je, aby tyto příkazy byly zpracovány dříve než první tabulka vytvořená tímto stylem v dokumentu. Lze je umístit také přímo v prostředí `tableenv` před definicí prvků

konkrétní tabulky. Toto umístění však není vhodné, protože při zblžném pohledu by příkazy na tomto místě mohly vzbuzovat dojem, že se týkají pouze tabulky, v jejímž prostředí jsou uvedeny, což není pravda. Mnohem lepší možnost představuje tyto příkazy umísťovat mimo prostředí `tableenv`, dokonce mohou být umístěny také před začátek samotného dokumentu (tedy před příkazem `\begin{document}`). Na tomto místě se již mohou vyskytovat jiná nastavení vztahující se k dokumentu.

6.2.3 Změna výchozího nastavení pro určitou tabulku v dokumentu

Druhou možností je, když se mají vzhledem odlišovat např. pouze některé tabulky v dokumentu. V takovém případě se příkazy mohou umístit před začátek prostředí `tableenv` patřící k příslušné tabulce nebo hned za začátek tohoto prostředí. Pokud však je těchto příkazů větší množství, nabízí se možnost umístit je do samostatného textového souboru a ten nechat na některém z uvedených umístění zpracovat. Pokud se však nejedná o poslední tabulku v dokumentu, je třeba si uvědomit, že pokud se neprovede nastavení zpět na původní hodnoty, budou tato nastavení aplikována také na všechny následující tabulky. Řešením může být opět jak uvedení příkazů, které původní hodnoty opět nastaví (pro lepší přehlednost a snadnou dohledatelnost např. ihned po ukončení prostředí nebo těsně před začátek prostředí patřící k následující tabulce), tak také používání dalšího souboru s původním nastavením. Pokud se však jednalo o poslední tabulku v dokumentu, není toto nastavení zpět na původní hodnoty nutné.

6.2.4 Změna výchozího nastavení pro část tabulky

Třetí možností, která může nastat, je, že uživatel potřebuje aplikovat změnu výchozích nastavení pouze na část tabulky. Pokud je touto částí jedna buňka, uživatel může použít lokální nastavení parametrů pomocí příslušných příkazů zapsaných ve volitelném parametru buňky. Pokud je touto oblastí jeden sloupec nebo větší část jednoho sloupce (nebo jedna mezisloupcová mezera, příp. její větší část), uživatel může využít některých implementovaných nastavení, která lze zadat při definici sloupce (nebo mezisloupcové mezery). V současné implementaci však druhů parametrů, které lze tímto způsobem zadat, je pouze několik. Pokud chce uživatel změnit jiný parametr, nemá jinou možnost než využít lokální nastavení každé buňky, které se odlišné formátování týká a všechny požadované parametry nastavit zde.

Pokud je požadovanou oblastí řádek, souvislá část řádku nebo přesněji řádkově souvislá oblast, je formátování výrazně jednodušší. Řádkově souvislou oblastí se rozumí část tabulky, která při čtení po řádcích od první buňky oblasti až po poslední buňku oblasti zahrnuje také všechny mezilehlé buňky. Příklady řádkově souvislých oblastí jsou ukázány na obr. 8.

Jednoduchost formátování řádkově souvislých oblastí vychází z toho, že na takové části tabulky lze použít změnu ve výchozím nastavení parametrů stejným způsobem, jaký byl popsán výše pro rozsah celé tabulky. Jen je v tomto případě nutné provést nastavení hodnot před definicí první buňky oblasti a nastavit hodnoty zpět

na původní hodnoty za definicí poslední buňky oblasti. Je ale třeba mít na paměti, jak už vyplývá z předchozích popisů situací, že změnu formátování na konci oblasti je nutné provést i v případě, že oblast končí až za poslední buňkou tabulky (výjimku tvoří opět situace, kdy se jedná o poslední tabulku v dokumentu). V opačném případě se změna formátování bude uplatňovat také na prvky stejného druhu i v následujících tabulkách a to až do chvíle, kdy bude výchozí nastavení změněno (nastavení lze samozřejmě provést např. před definicí následující tabulky, v tomto speciálním případě tedy není nutné provést nastavení ihned po ukončení oblasti) nebo do konce dokumentu.

Obr. 8: Ukázky různých řádkově souvislých oblastí.

Tento způsob formátování je samozřejmě vhodný i v případě, že požadovaná oblast formátování zcela souvislá není. V takovém případě lze formátování používat opakovaně (na menší, ale již zcela souvislé oblasti), příp. může být výhodnější buňky, které způsobují nesouvislost oblasti, odlišně formátovat lokálně. Tento způsob formátování lze samozřejmě použít i pro jedinou buňku, i ta sama splňuje podmínku řádkově souvislé oblasti, ale již popsáný způsob lokálního nastavení představuje mnohem pohodlnější možnost. V případě dvou buněk vyžadují oba způsoby dvě změny (lokální formátování první i druhé buňky a nebo změna výchozího formátování před první buňkou a za druhou buňkou). V případě oblasti o rozsahu tří buněk je již způsob formátování pomocí změny výchozího nastavení parametrů výhodnější.

Pokud uživatel používá při formátování (nezáleží, zda se jedná o lokální formátování nebo formátování výchozích hodnot) více příkazů, může být užitečné definovat jeden příkaz, který všechny změny provede. Velmi výhodné je tuto metodu použít, pokud takové formátování provádí uživatel opakovaně (např. stejná změna lokálního formátování více nesouvislých buněk nebo stejné formátování více řádkově souvislých oblastí). Pro formátování řádkově souvislé oblasti může mít definice takového příkazu následující tvar:

```
\def\defprikazABC{\defprikazA{0}\defprikazB{.5}\defprikazC{1}}
```

Takový příkaz pak může být uváděn před každou požadovanou formátovanou oblastí. Dále je samozřejmě ještě vhodné definovat příkaz, který nastaví modifikované parametry zpět na původní hodnotu a uvádět jej vždy za požadovanou formátovanou oblastí. Může se tak dosáhnout zpřehlednění textu a např. také změny formátování v mnoha řádkově souvislých oblastech s požadavkem na stejné formátování budou vyžadovat provedení změn pouze na jediném místě.

6.2.5 Definice sloupců, meziřádkových mezer a příp. uvedení titulku tabulky

Definici sloupců a meziřádkových mezer je již třeba umístit do prostředí `tableenv` patřící k příslušné tabulce. Pokud zde nejsou příkazy modifikující výchozí nastavení parametrů, měl by to být první zápis, který bude toto prostředí obsahovat. Zápis se provádí tak, že jsou střídavě definovány sloupce a mezisloupcové mezery, dokud není definován požadovaný počet sloupců. Tento zápis musí začínat i končit definicí mezisloupcové mezery. Počet definovaných mezisloupcových mezer tedy bude vždy o jednu vyšší než počet definovaných sloupců. V současné implementaci je takový zápis nutný i v případě, že žádné mezisloupcové mezery nejsou požadovány. V takovém případě je nutné zapsat mezisloupcovou mezeru s nulovou šířkou.

Příkaz pro definici sloupce je `\col` a má jeden povinný parametr, kterým je výchozí horizontální zarovnání buněk v daném sloupci. Dále má jeden volitelný parametr, který určuje jeho šířku.

Příkaz pro definici mezisloupcové mezery je `\colspace` bez povinného parametru, ale s jedním volitelným parametrem. Co může být obsahem povinného parametru příkazu pro definici sloupce a volitelných parametrů obou příkazů je uvedeno dále v této kapitole u popisu implementace těchto příkazů. Jednoduchá definice tabulky o třech sloupcích (všech s výchozím horizontálním zarovnáním vlevo) bez použití volitelných parametrů je následující:

```
\colspace
\col{1}
\colspace
\col{1}
\colspace
\col{1}
\colspace
```

Definice každého prvku samozřejmě nemusí být provedena vždy na nový řádek, ale minimálně v případech, kdy uživatel často využívá volitelných parametrů, je takovýto zápis přehlednější.

Po této definici může následovat definice titulku tabulky (pokud je titulek požadován). Jeho definici je nutné provést až po definici sloupců a mezisloupcových

mezer, protože se šířka parboxu pro umístění titulku vypočítává podle šířky tabulky, jak bude ještě uvedeno dále u popisu implementace titulku tabulky, kde jsou také uvedeny možnosti, jak titulek formátovat.

Za definicí sloupců, sloupcových mezer a příp. titulku tabulky (pokud zde nejsou umístěné již zmíněné příkazy pro změnu výchozího nastavení) již následuje definice samotných buněk tabulky.

6.2.6 Definice buněk

Pro sazbu buňky slouží příkaz `\cell` s jedním povinným a jedním volitelným parametrem. Povinný parametr určuje obsah buňky. Rovněž se používá k formátování textu uvnitř buňky. Buňky jsou vysazovány vedle sebe na řádku až do doby, kdy je uveden příkaz `\rowend`. Počet použitých sloupců na řádku (nemusí být vždy stejný jako počet buněk na řádku, vezmou-li se v úvahu buňky sloučené přes více sloupců, jejichž implementace je také zahrnuta) nesmí být vyšší než počet definovaných sloupců. Poté pokračuje sazba dalších buněk na začátku následujícího řádku. Příkaz `\rowend` má také jeden volitelný parametr. U popisu jeho implementace dále v této kapitole je uvedeno, co může obsahovat. Při sazbě je však třeba vědět, že v současné implementaci musí být tento příkaz použit i před definicí prvního a za definicí posledního řádku. Důvody jsou podobné jako v případě nutnosti používat příkaz `\colspace` před definicí prvního a za definicí posledního sloupce. Jak je buňka implementována a co vše může být uvedeno ve volitelném parametru příkazu pro její sazbu je detailně popsáno dále v této kapitole. Pro názornost je ještě možné na příkladě ukázat definici buněk jednořádkové tabulky o třech sloupcích (bylo by tedy možné ji využít ve spojení s předchozím příkladem, ve kterém byly definovány tři sloupce a příslušné mezisloupcové mezery):

```
\rowend
\cell{A}
\cell{B}
\cell{C}
\rowend
```

Opět není nutné psát definici každé buňky vždy na nový řádek, ale při užívání příkazů lokálního nastavení (kterých je k dispozici velké množství), se může jednat o přehlednější způsob zápisu.

Takto tedy lze ve stručnosti shrnout popis práce se stylem. Případná sazba poznámek pod tabulkou (pokud jsou v tabulce použity) probíhá automaticky podle jejich nastavení, ale také tyto poznámky lze formátovat pomocí jejich výchozího nastavení.

6.2.7 Definované délkové registry a jejich využití

Ve stylu je použito celkem devatenáct délkových registrů. Kromě dvou nejdůležitějších (`\coordinateX` a `\coordinateY`), které se používají nejčastěji, se používá ještě dalších sedmnáct délkových registrů k různým účelům. Mnohem větší množství délek je ale užíváno ve formě příkazů. Uživatel nemá důvod do hodnot v délkových registrech nijak zasahovat. Všechny použité délkové registry včetně jejich účelu jsou uvedeny v tab. 3.

Tab. 3: Seznam definovaných délkových registrů a jejich využití

Název registru	Použití registru
<code>\coordinateX</code>	souřadnice sazby X
<code>\coordinateY</code>	souřadnice sazby Y
<code>\rowheight</code>	výška řádku
<code>\cellwidth</code>	šířka buňky
<code>\cellcontentheight</code>	výška obsahu buňky
<code>\cellcontentwidth</code>	šířka obsahu buňky
<code>\typetextheight</code>	výška textu v tabulce po výpočtu
<code>\typetextdepth</code>	hloubka textu v tabulce po výpočtu
<code>\typetextheightdepth</code>	součet výšky a hloubky textu v tabulce po výpočtu
<code>\titletextheight</code>	výška titulku po výpočtu
<code>\titletextdepth</code>	hloubka titulku po výpočtu
<code>\titletextheightdepth</code>	součet výšky a hloubky titulku po výpočtu
<code>\rowpaceprev</code>	výška předchozí meziřádkové mezery
<code>\rowpacenext</code>	výška následující meziřádkové mezery
<code>\colspaceprev</code>	šířka předchozí mezisloupcové mezery
<code>\colspacenext</code>	šířka následující mezisloupcové mezery
<code>\linehfirstcol</code>	přetah horizontálních linek v prvním sloupci doleva
<code>\linehlastcol</code>	přetah horizontálních linek v posledním sloupci doprava
<code>\tablewidth</code>	šířka tabulky

6.2.8 Definované čítače a jejich využití

Čítačů je ve stylu použito celkem čtrnáct. Tak jako do hodnot v délkových registrech, ani do hodnot v čítačích uživatel nepotřebuje zasahovat. Sice by některé operace mohl provádět přímou manipulací s jejími hodnotami, ale pro běžné operace existuje vždy taková možnost, aby se uživatel přímé manipulaci s hodnotami čítačů mohl vyhnout, ale ve speciálních případech to uživatel může potřebovat. Hodnoty některých z nich jsou změněny hned po jejich definici, některých předtím, než se s nimi začne pracovat. Podrobnosti jsou uvedeny v této kapitole u popisu imple-

mentace příslušných prvků, které dané čítače používají. Seznam všech čítačů a jejich použití je uveden v tab. 4.

Tab. 4: Seznam definovaných čítačů a jejich využití

Název čítače	Použití čítače
<code>\column</code>	pořadí sloupce
<code>\row</code>	pořadí řádku
<code>\merged</code>	počet sloučených buněk
<code>\tablecount</code>	pořadí tabulky
<code>\columncount</code>	počet sloupců
<code>\cellcount</code>	počet započítaných sloupců
<code>\lastcolumn</code>	pořadí posledního sloupce
<code>\tablenotecount</code>	počet poznámek pod tabulkou
<code>\tablenumcount</code>	počet automaticky číslovaných poznámek pod tabulkou
<code>\note</code>	pořadí poznámky pod tabulkou
<code>\cellmark</code>	pořadí poznámky pod tabulkou v buňce
<code>\cellmarkcount</code>	počet značek poznámek pod tabulkou v buňce
<code>\linestyleleft</code>	styl levé linky buňky
<code>\linestylerright</code>	styl pravé linky buňky

6.2.9 Definované struktury a jejich využití

Strukturami se v pojetí těch, které se používají v tomto stylu, rozumí jednorozměrná dynamická pole. Ve stylu se jich používá celkem osm, i když v části definice struktur, kterou lze nalézt v souboru `tableenvironment.sty`, jich je definováno pouze sedm. Těchto sedm struktur se používá v rámci celé tabulky. Osmá struktura (`\cellmarks`) je definována až v příkazu pro sazbu buněk (již zmíněný příkaz `\cell`). Tato struktura se (jako jediná) používá v rámci každé buňky vždy znovu. Seznam definovaných struktur a jejich použití je uveden v tab. 5.

Tab. 5: Seznam definovaných struktur a jejich využití

Název struktury	Použití struktury
<code>\rowheights</code>	uchování výšky jednotlivých řádků v tabulce
<code>\colwidths</code>	uchování šířky jednotlivých sloupců v tabulce
<code>\collinestyles</code>	uchování výchozích stylů linek mezi sloupci
<code>\colaligns</code>	uchování výchozích horizontálních zarovnání obsahu buněk ve sloupcích
<code>\colspaces</code>	uchování šířek mezisloupcových mezer
<code>\marks</code>	uchování značek poznámek pod tabulkou
<code>\notes</code>	uchování textů poznámek pod tabulkou
<code>\cellmarks</code>	uchování značek poznámek pod tabulkou pro uvedení v buňce

Nejprve je třeba strukturu definovat:

```
\def\strukturaA{"a"}
```

Touto definicí již struktura získává svůj první prvek (`a`). Pak lze do těchto struktur přidávat nové prvky následujícím způsobem:

```
\edef\strukturaA{\strukturaA,"b"}
```

`\strukturaA` teď tedy obsahuje dva prvky, `a` a `b`. Všechny ve stylu použité struktury jsou tedy s jedním prvkem již definovány. Tímto prvkem je velké písmeno anglické abecedy. Použita byla písmena `A–H` (pro každou strukturu jiné písmeno). Obsah prvního prvku každé struktury ale není důležitý, nikdy se k němu totiž nepřistupuje. Struktura s jedním prvkem je tedy vzhledem k použití v tomto stylu vlastně považována za prázdnou. Důležité jsou až prvky následující (pokud takové jsou). Jak přistupovat k těmto prvkům je popsáno v (Tantau aj., 2010).

6.3 Prostředí `tableenv`

Základní definicí stylu je definice samotného prostředí `tableenv` (table environment). Všechny příkazy související s tvorbou tabulky v tomto prostředí musí být umístěny mezi příkazy pro začátek `\begin{tableenv}` a konec `\end{tableenv}` tohoto prostředí (výjimka v podobě příkazů pro změnu výchozích nastavení již byla detailně rozvedena výše). Prostředí `tableenv` je definováno následovně:

```
\newenvironment{tableenv}{\vspace{\@spacevtop}\noindent
\begin{tikzpicture}}{\tablenotes\end{tikzpicture}
\vspace{\@spacevbottom}}
```

Základem této definice je to, že obsah prostředí `tableenv` je současně umístěn v prostředí prostředí `tikzpicture`, ve kterém je řešena sazba všech prvků tabulky.

Není to ale to jediné, co definice prostředí `tableenv` zajišťuje. Jak je vidět v jeho definici, pro umístění tabulky odstraní případné horizontální odsazení odstavce a na konci tabulky zajistí vysazení poznámek pod tabulkou. Implementace všech prvků zajišťujících sazbu poznámek pod tabulkou je popsána dále v této kapitole.

Definice tohoto prostředí také používá dva nové příkazy, které jsou příkladem příkazů pro změnu výchozího nastavení, oba s jedním povinným parametrem:

- příkaz `\spacevtop` definuje vertikální mezeru nad tabulkou,
- příkaz `\spacevbottom` definuje vertikální mezeru pod tabulkou.

Obě výchozí hodnoty těchto vertikálních mezer jsou nastaveny na `0mm`. Podle tohoto nastavení je tedy vertikální mezeru nad tabulku i pod ní nulová.

Takto je možné např. nastavit obě mezery na velikost 2 cm:

```
\spacevtop{2cm}
\spacevbottom{2cm}
```

Pokud zapíšeme text dvou dříve zmíněných příkladů a sice příklad definice sloupců a mezisloupcových mezer a příklad jednořádkové tabulky o třech sloupcích (v tomto pořadí) mezi příkazy `\begin{tableenv}` a `\end{tableenv}`, získáme jednoduchou tabulku:

Tab. 6: Příklad jednoduché tabulky vytvořené již v prostředí `tableenv`.

A	B	C
---	---	---

6.4 Používání vrstev

Používání sazby v různých vrstvách je velmi užitečným nástrojem. Především je s jeho pomocí řešen jeden z hlavních požadavků na styl a sice umožnit nastavení překrývání křížících se linek. Byl ale zvolen komplexní přístup a vrstvy jsou využívány v celé sazbě.

V PGF se na vrstvy odkazuje pomocí jmen. Standardní vrstva, která je v určitých směrech trochu speciální, se nazývá `main`. Když není nic dalšího specifikováno, všechny grafické příkazy jsou připojeny do vrstvy `main`. (Tantau aj., 2010)

Ve stylu však do této vrstvy nejsou přiřazeny žádné prvky. Další vrstvy jsou ve stylu definovány čtyři. Do nich byly ve výchozím nastavení rozděleny všechny prvky tabulky. Jak se nové vrstvy deklarují a jak se používají je uvedeno v (Tantau aj., 2010).

Dalším krokem směrem k použití vrstvy je říci PGF které vrstvy budou součástí aktuálního obrázku a jak budou řazeny. Takto je možné mít deklarováno více vrstev, než je aktuálně používáno. (Tantau aj., 2010)

Způsob, jak se jednotlivé vrstvy vrší na sebe je rovněž popsán v (Tantau aj., 2010). Definované čtyři vrstvy jsou pojmenovány `layerone`–`layerfour`. Rozdělení prvků tabulky do vrstev je uvedeno v tab. 7.

Tab. 7: Výchozí rozmístění prvků tabulky do vrstev

pořadí	název vrstvy	prvky ve vrstvě podle výchozího nastavení
1.	<code>layerone</code>	pozadí buňky, rámeček buňky, přeškrtnutí buňky
2.	<code>layertwo</code>	levá linka buňky, pravá linka buňky
3.	<code>layerthree</code>	horní linka buňky, dolní linka buňky
4.	<code>layerfour</code>	obsah buňky, titulek tabulky, poznámky pod tabulkou včetně oddělovací linky
5.	<code>main</code>	žádné

Pokud by pro uživatele nebyl současný počet vrstev dostatečný, může samozřejmě zasáhnout do souboru `tableenvironment.sty` a tam definovat jednu nebo více dalších vrstev a některé prvky tabulky do nich přesunout.

Takto je tedy vyřešena možnost nastavení překrývání křížících se linek. Podle výchozího nastavení jsou horizontální (horní a dolní) linky ve třetí vrstvě (`layerthree`) a vertikální (levá a pravá) linky ve druhé vrstvě (`layertwo`). Podle tohoto nastavení jsou horizontální linky ve vyšší vrstvě a překryjí tedy linky vertikální.

Pokud má některá vertikální linka překrýt linku horizontální, stačí ji přesunout do vyšší vrstvy, než ve které je umístěna horizontální linka, se kterou se kříží. Např. je tedy možné takovouto vertikální linku přesunout do čtvrté vrstvy (`layerfour`), která je již sice obsazena obsahem buňky, titulkem tabulky a poznámkami pod tabulkou, ale to by většinou nemělo vadit, protože s těmito prvky by linka neměla v běžných případech přijít do kontaktu. Případně vždy existuje možnost nadefinovat si pro takovéto vertikální linky novou vrstvu (a umístit ji na patřičné místo) nebo použít již zmíněnou vrstvu `main`.

Zarážející může být umístění pozadí buňky, rámečku buňky a přeškrtnutí buňky podle výchozího nastavení do stejné vrstvy, ale toto společné umístění potíže nezpůsobuje.

Vrstva `main` je zařazena jako nejvyšší vrstva. Je to z toho důvodu, že pokud uživatel bude do prostoru tabulky cokoliv umísťovat, pravděpodobně bude požadovat (opět jde o možnost, které by mohli využívat pravděpodobně pouze zkušení uživatelé a pouze ve výjimečných případech), aby tento objekt byl viditelný.

Aby bylo možné měnit rozmístění prvků ve vrstvách, jsou definovány pro každý prvek jeden až dva příkazy, které přesuny do jiné vrstvy zajišťují. Pro titulek tabulky a poznámky pod tabulkou je definován pouze příkaz pro výchozí změnu, ne lokální. Lze totiž předpokládat, že toto nastavení bude stejné pro všechny tabulky v dokumentu. Pro všechny ostatní prvky existují obě verze příkazů. Jejich seznam uvádí tab. 8.

Tab. 8: Seznam příkazů pro přesun prvků tabulky do jiné vrstvy

prvek tabulky	lokální změna vrstvy	výchozí změna vrstvy
obsah buňky	<code>\cellcontentlayer</code>	<code>\defcellcontentlayer</code>
titulek tabulky	příkaz není definován	<code>\titlelayer</code>
poznámky pod tabulkou včetně oddělovací linky	příkaz není definován	<code>\notelayer</code>
horní linka buňky	<code>\linetoplayer</code>	<code>\deflinetoplayer</code>
levá linka buňky	<code>\lineleftlayer</code>	<code>\deflineleftlayer</code>
dolní linka buňky	<code>\linebottomlayer</code>	<code>\deflinebottomlayer</code>
pravá linka buňky	<code>\linerightlayer</code>	<code>\deflinerightlayer</code>
pozadí buňky	<code>\blayer</code>	<code>\defblayer</code>
rámeček buňky	<code>\framelayer</code>	<code>\defframelayer</code>
přeškrtnutí buňky	<code>\strokelayer</code>	<code>\defstrokelayer</code>

Všechny uvedené příkazy mají jeden povinný parametr, který určuje název cílové vrstvy. Pokud by uživatel např. chtěl nastavit výchozí zařazení vertikálních linek do čtvrté vrstvy (`layerfour`), mohl by to zajistit následujícími dvěma příkazy:

```
\deflineleftlayer{layerfour}
\deflinerightlayer{layerfour}
```

Takové řešení je možné, ale je na něj nutné použít dva příkazy. V případě linek lze očekávat, že to budou prvky tabulky, které se do jiných vrstev budou přesouvat nejčastěji. Proto jsou součástí implementace také čtyři další příkazy, kterými lze obě horizontální nebo obě vertikální linky přesunout do jiné vrstvy najednou. Tyto příkazy opět mají jeden povinný parametr, který určuje název cílové vrstvy. Jejich seznam je uveden v tab. 9.

Tab. 9: Seznam příkazů pro přesun horizontálních nebo vertikálních linek do jiné vrstvy

prvky tabulky	lokální změna vrstvy	výchozí změna vrstvy
horizontální linky	<code>\lineshlayer</code>	<code>\deflineshlayer</code>
vertikální linky	<code>\linesvlayer</code>	<code>\deflinesvlayer</code>

Uživateli tedy k výše uvedenému přesunu do jiné vrstvy stačí pro obě vertikální linky použít (narozdíl od předchozího příkladu) pouze jeden příkaz:

```
\deflineshlayer{layerfour}
```

V případě, že má při křížení kterákoliv vertikální linka překrýt kteroukoliv horizontální linku, je možné jejich vrstvy vzájemně zaměnit. K tomuto účelu je rovněž použití příkazů pro přesun obou horizontálních a obou vertikálních linek vhodnější než použití čtyř samostatných příkazů pro přesun horní, dolní, levé a pravé linky.

Jak vrstvy používat je uvedeno v (Tantau aj., 2010). Sazba každého prvku tabulky probíhá v příslušné vrstvě, což v popisu těchto prvků v této kapitole není uváděno.

6.5 Slovně pojmenované délky

Slovně pojmenované délky jsou dvě jednoduché, ale velmi užitečné definice. Uživatel jejich využitím získává možnost používat dvě (různé) slovně vyjádřené délky místo přímého číselného zápisu.

Výhody jsou zřejmé. Uživatel tyto délky nadefinuje jednou a pak už si konkrétní rozměry nemusí pamatovat, protože se na ně odkazuje slovně. Navíc se poté velmi jednoduše provádí změny těchto délek. Uživatel je změní jednou a tím se automaticky změní tyto délky všude tam, kde slovně pojmenovanou délku použil. Využití se

předpokládá především pro definici šířek linek a mezisloupcových a meziřádkových mezer. Délky se jmenují následovně:

- `\thin` (tenká)
- `\thick` (silná)

Výhodou dále je, že jedna z těchto délek může být nadefinována závisle na druhé. Tak je tomu i v jejich výchozím nastavení. Délka `\thin` je nastavena na `1pt`, délka `\thick` je nastavena na `2*\thin`. Pro změnu každé z těchto délek je definován příkaz. Jsou to:

- `\thindf` pro definici délky `\thin`,
- `\thickdf` pro definici délky `\thick`.

Oba příkazy mají jeden povinný parametr, který určuje novou délku. Je možné je uvést přímo v dokumentu nebo je lze nalézt v souboru s výchozím nastavením (`tableenvdefaults.sty`).

6.6 Sloupec a mezisloupcová mezera

Základy používání příkazů `\col` a `\colspace` již byly uvedeny výše. Bylo řečeno jakým způsobem se tyto příkazy používají v definici tabulky. Teď bude uvedeno zejména jak jsou tyto příkazy implementovány. Důležité je, že každý sloupec zleva i zprava sousedí s mezisloupcovou mezerou a to včetně okrajových sloupců.

6.6.1 Sloupec

Již bylo uvedeno, že příkaz `\col`, který pro definici sloupce slouží, má dva parametry, jeden povinný a jeden volitelný. Povinný parametr určuje výchozí horizontální zarovnání obsahu buněk v daném sloupci. K dispozici jsou tři běžně užívané možnosti zarovnání. Do parametru se vkládá počáteční písmeno názvu tohoto zarovnání. Tato písmena byla přejata z prostředí `tabular`, kde se používají pro totéž. Jsou popsána v (Rybička, 2003).

- `l` pro zarovnání vlevo,
- `c` pro zarovnání na střed,
- `r` pro zarovnání vpravo.

Volitelný parametr určuje šířku sloupce. Zde se nachází pravděpodobně největší nedostatek tohoto stylu. Pokud tento parametr není v definici sloupce uveden, je použita výchozí šířka sloupce (`\@defcolwidth`), která je součástí výchozího nastavení a lze ji změnit příkazem `\defcolwidth` s jedním povinným parametrem, který výchozí šířku sloupce určuje a ve výchozím nastavení je tato šířka nastavena na `2cm`. To však není optimální, protože v takovém případě by se měla šířka sloupce určovat podle nejširšího obsahu buňky v daném sloupci. To je běžné chování nejen v prostředí `tabular`. Tuto funkcionalitu však styl v jeho současné implementaci neumožňuje.

Sloupec je implementován následujícím způsobem. Nejprve do struktury `\colwidths` (určené pro uchovávání šířek sloupců) přidá obsah svého volitelného parametru. Pokud chybí, přidá výchozí šířku sloupce. Poté do struktury `\colaligns`

(určené pro uchovávání výchozího horizontálního zarovnání sloupců) přidá obsah svého povinného parametru. Poté inkrementuje čítač `\columncount` (určený pro uchovávání počtu sloupců). Nakonec ještě šířku právě přidaného sloupce přičte k hodnotě délkového registru `\tabwidth` (určeného pro uchovávání celkové šířky tabulky).

Například sloupec o šířce 5 cm s výchozím horizontálním zarovnáním na střed lze definovat takto:

```
\col[5cm]{c}
```

6.6.2 Mezisloupcová mezera

Také základy příkazu `\colspace`, který slouží pro definici mezisloupcové mezery, již byly zmíněny. Bylo uvedeno, že se jedná o příkaz s jedním volitelným parametrem. V něm může být uveden libovolný počet ze dvou následujících příkazů. Jedná se o příkazy s jedním povinným parametrem, který definuje dané nastavení:

- `\colspacewidth` definuje šířku mezisloupcové mezery. Není-li tento příkaz použit, je použita výchozí šířka mezisloupcové mezery `\defcolwidth`, ve výchozím nastavení je rovna délce `\thin`.
- `\collinestyle` definuje styl linky v mezisloupcové mezeře. Není-li tento příkaz použit, je použit výchozí styl linky `\deflineleftstyle` nebo `\deflinerightstyle` (který z nich se použije je uvedeno dále). Práce se styly čar je popsána v (Tantau aj., 2010). V současné implementaci stylu jsou zahrnuty tři styly linek. Parametr příkazu se zapisuje jako jedno z čísel:
 - 1 pro plnou linku
 - 2 pro tečkovanou linku
 - 3 pro čárkovanou linku

Výchozí nastavení zmíněných parametrů lze samozřejmě změnit. Slouží k tomu následující příkazy (každý z nich má jeden povinný parametr určující nové výchozí nastavení):

- `\defcolspacewidth` určuje výchozí šířku mezisloupcové mezery (výchozí nastavení je `\thin`)
- `\deflineleftstyle` určuje výchozí styl levé linky (výchozí nastavení je 1)
- `\deflinerightstyle` určuje výchozí styl pravé linky (výchozí nastavení je 1)

Například druhou mezisloupcovou mezeru o šířce 5 pt s výchozím stylem tečkované linky lze definovat takto:

```
\colspace[\colspacewidth{5pt}\collinestyle{2}]
```

Příkaz mezisloupcové mezery nejdříve nastaví příkazy pro lokální změnu nastavení `\colspacewidth` a `\collinestyle` na hodnoty příslušných příkazů pro výchozí nastavení. Pro výchozí nastavení šířky mezisloupcové mezery je to

`\defcolspacewidth` a pro výchozí styl linky je to `\deflineleftstyle` v případě, že se jedná o první mezisloupcovou mezeru (vlevo od prvního sloupce) a `\deflinerightstyle` pokud se jedná o kteroukoliv následující mezisloupcovou mezeru.

Následně se provedou změny prostřednictvím příkazů uvedených ve volitelném parametru. Případné uvedení jednoho nebo obou těchto příkazů způsobí změnu v lokálním nastavení. Následně se do struktury `\colspaces` (určené pro uchovávání šířek mezisloupcových mezer) přidá aktuální hodnota `\@colspacewidth` a do struktury `\collinestyles` se přidá aktuální hodnota `\@collinestyle`. Samozřejmě také šířky mezisloupcových mezer je nutné započítat do celkové šířky tabulky, takže i šířka právě přidané mezisloupcové mezery je připočtena k hodnotě délkového registru `\tabwidth`.

6.7 Řádek a meziřádková mezera

Řádek tabulky tvoří samotné buňky, které budou popsány dále v této kapitole. Každý řádek shora i zdola sousedí s meziřádkovou mezerou, což platí také pro první a poslední řádek.

6.7.1 Ukončení řádku

Jak již bylo uvedeno, pro ukončení řádku je definován příkaz `\rowend`. Tento příkaz slouží nejen k ukončování aktuálního řádku, ale lze jej využít také k formátování řádku následujícího včetně jeho dolní meziřádkové mezery. K tomu slouží volitelný parametr tohoto příkazu, do kterého lze zapsat libovolný počet z následujících dvou příkazů. Oba tyto příkazy mají jeden povinný parametr nastavující daný parametr:

- `\nextrowheight` definuje výšku následujícího řádku. Není-li tento příkaz použit, je použita výchozí výška řádku, která je ve výchozím nastavení nastavena na hodnotu 0, což ale neznačí nulovou výšku řádku. Při této hodnotě se výška řádku určuje z výchozího stupně písma, což bude ještě popsáno dále. Stejně jako automatická volba šířky sloupce, není ve stylu implementována ani automatická výška řádku podle nejvyššího obsahu buňky v něm umístěné. Výchozí výšku řádků lze případně změnit příkazem `\defrowheight`
- `\nextrowspace` definuje šířku dolní meziřádkové mezery následujícího řádku (nejedná se tedy o meziřádkovou mezeru pod řádkem, který je příkazem `\rowend` právě ukončován). Výchozí výška je dána příkazem `\defrowspace` a ve výchozím nastavení je tato výška rovna pojmenované délce `\thin` stejně jako šířka mezisloupcové mezery.

Již bylo také uvedeno, že tento příkaz je nutné použít také před definicí prvního řádku a také za definicí posledního řádku. Před prvním řádkem nastavuje vlastnosti právě prvního řádku. Pokud ale tento příkaz nastavuje až dolní meziřádkovou mezeru následujícího řádku, jednu z meziřádkových mezer nelze pomocí tohoto příkazu nastavit a sice první, tedy meziřádkovou mezeru nad prvním řádkem.

K tomuto účelu slouží příkaz `\firstrowspace`. Je velmi podobný příkazu `\defrowspace`, ale platí pouze pro první meziřádkovou mezeru. Jeho výchozí hodnota je nastavena na `\@defrowspace`, což bez dalších změn v šířkách meziřádkových mezer znamená, že první mezera bude mít stejnou šířku jako všechny ostatní meziřádkové mezery.

Pro příklad byla zvolena definice ukončení řádku, která nastaví výšku následujícího řádku na 1 cm a šířku po něm následující meziřádkové mezery na 10 pt:

```
\rowend[\nextrowheight{1cm}\nextrowspace{10pt}]
```

Příkaz pracuje tak, že nejprve nastaví lokální hodnoty nastavení následující meziřádkové mezery a výchozí šířky řádku na hodnoty výchozích nastavení. Poté se případně provedou změny v těchto hodnotách, pokud jsou použity příslušné příkazy ve volitelném parametru. Poté se délkový registr souřadnice X (`\coordinateX`) nastaví na začátek řádku, což odpovídá hodnotě první meziřádkové mezery. Také souřadnici Y je nutné nastavit. Od současné hodnoty uložené v délkovém registru pro tuto souřadnici (`\coordinateY`) se odečte šířka meziřádkové mezery pod tímto řádkem (v případě nastavení na první řádek šířka první mezisloupkové mezery `\firstrowspace`). V případě, že se jedná o nastavení na jiný řádek než první, odečte se také výška aktuálního řádku.

Dále se nastaví hodnoty do délkových registrů pro uchovávání šířky předchozí (`\rowSPACEprev`) a následující meziřádkové mezery (`\rowSPACEnext`). Při přechodu na další řádek samozřejmě přestane platit dosavadní hodnota následující meziřádkové mezery a tato hodnota se stává hodnotou předchozí meziřádkové mezery a tato hodnota se nastaví do délkového registru `\rowSPACEprev`. Šířka následující meziřádkové mezery se získá z příkazu ve volitelném parametru a pokud ten není uveden, nastaví se hodnota výchozí meziřádkové mezery. Nová hodnota se nastaví do délkového registru `\rowSPACEnext`.

Pak se výška následujícího řádku přidá do struktury `\rowheights`, která k uchovávání výšky řádků slouží.

Nakonec se ještě čítač pro uchovávání pořadí sloupce `\column` nastaví zpět na první sloupec (na hodnotu 1) a inkrementuje se čítač pro uchovávání pořadí řádku `\row`.

6.8 Výchozí stupeň písma v různých prvcích tabulky

Styl nabízí také možnost definovat výchozí stupeň písma v různých prvcích tabulky ve formě příkazů pro výchozí nastavení s jedním povinným parametrem definujícím výchozí stupeň písma při sazbě daného prvku:

- `\typesize` pro stupeň písma použitým v buňkách tabulky (výchozí nastavení je `\small`)
- `\titlesize` pro stupeň písma titulku tabulky (výchozí nastavení je `\normalsize`)

- `\notesize` pro stupeň písma poznámek pod tabulkou (výchozí nastavení je `\footnotesize`)

Chce-li uživatel např. zvětšit stupeň písma poznámek pod tabulkou, může to provést zápisem následujícího příkazu:

```
\notesize{\small}
```

Nastavení stupně písma je v případě stupně písma v buňkách a titulku tabulky používáno ještě pro dvě další operace:

- v případě buněk když je výška řádku nastavena na 0 (tedy že se má použít výchozí výška řádku) k výpočtu celkové výšky buňky. Výpočet je uveden dále v této kapitole. V případě titulku tabulky se používá vždy a to k posunu souřadnice Y dolů po sazbě nadpisu. Počítá se s jednořádkovým titulkem. V případě, že bude titulek přesahovat na více řádků, je třeba nastavit vyšší vzdálenost titulku od vlastní tabulky. Příkaz, který tuto vzdálenost nastavuje, je uveden dále. Tento příkaz byl určen spíše pro použití s jednořádkovými titulky, ale tento problém při víceřádkovém titulkem lze s jeho pomocí řešit
- k částečné kompenzaci vertikálního zarovnání textu v parboxech podle okraje textu a ne podle účarí zapsáním strutu popsáném v (Rybička, 2003) před parametrem definujícím obsah buňky (nebo titulku) a také za ním

V obou případech se využívá výška a hloubka řetězce (Ěj. Funkce pro zjištění výšky a hloubky boxu jsou popsány v (Tantau aj., 2010).

6.9 Titulek tabulky

O titulkem tabulky již bylo uvedeno, že je třeba jej definovat až po definici všech sloupců a mezisloupcových mezer. Je to z toho důvodu, že titulek je horizontálně zarovnáván vzhledem k tabulce. Šířka parboxu pro umístění titulku je stejná jako šířka tabulky. Představuje to tedy jedno z využití celkové šířky tabulky uchovávané v délkovém registru `\tablewidth`. Titulek je umístěn vertikálně nad tabulkou.

Pro definici titulku tabulky slouží příkaz `\title` s jedním povinným parametrem, který titulek určuje. Změna výchozího stupně písma titulkem již byla popsána výše. Je ale možnost formátovat ještě jiné parametry titulku:

- příkazem `\titlealign` horizontální zarovnání titulku v rámci šířky tabulky, možná zarovnání jsou stejná jako pro horizontální zarovnání buňky, která byla přejata z prostředí `tabular`, výchozí horizontální zarovnání je 1,
- příkazem `\titlespace` pro nastavení šířky vertikální mezery mezi titulkem tabulky a okrajem první meziřádkové mezery tabulky, výchozí hodnota je 0mm, také lze s jeho pomocí řešit výše zmíněný problém víceřádkového titulku,
- příkazem `\tablenumbering` pro zapnutí nebo vypnutí inkrementace čítače pořadí tabulky při použití titulku (inkrementace se provádí před jeho sazbu), výchozí hodnota je 1 (zapnuto),

- je možné předefinovat příkaz `\tabletitle` určující automatický text v titulku (celý text titulku kromě názvu samotné tabulky, který je definován v povinném parametru příkazu `\title`) a tak tento automatický text změnit. Tento příkaz je ještě popsán dále.

Všechny uvedené příkazy mají jeden povinný parametr určující dané nastavení.

Již uvedený příklad jednoduché tabulky lze rozšířit např. o následující definici titulku (na příslušném umístění):

```
\title{Tabulka s titulkem}
```

Tabulka bude poté vypadat takto:

Tab. 10: Příklad jednoduché tabulky s titulkem vytvořené v prostředí `tableenv`.

Tab. 1: Tabulka s titulkem

A	B	C
---	---	---

Výchozí definice příkazu `\tabletitle` je následující:

```
\def\tabletitle#1{Tab. \tablenumber : #1}
```

Příkaz `\tablenumber` zajišťuje sazbu čísla pořadí tabulky, jeho definice je následující:

```
\def\tablenumber{\the\tablecount}
```

Tento příkaz tedy jen zkracuje původní zápis. Samozřejmě je možné ho v definici používat. Může být použit při předefinování, ale také nemusí, pokud součástí titulku pořadí tabulky být nemá. Podle této definice tedy titulek vypadá tak, že začíná Tab., následuje její číslo, po něm :, mezera a poté název tabulky určený povinným parametrem příkazu `\title`.

Uživatel může předefinovat přímo původní definici umístěnou v souboru `tableenvironment.sty` (ne v souboru `tableenvdefaults.sty`, jedná se o výchozí definici, ne o výchozí nastavení, i když zda tuto definici přesunout do souboru s výchozím nastavením je na zvážení) nebo uvést novou definici stejně, jako by se jednalo o změnu výchozího nastavení. Tato definice původní definici v souboru `tableenvironment.sty` nahradí.

6.9.1 Implementace příkazu

Příkaz nejprve inkrementuje hodnotu čítače `\tablecount`, pokud je tato inkrementace zapnutá. Poté je na základě aktuální (jednoznakové) hodnoty v `\@titlealign` přiřazen příkaz, který dané horizontální zarovnání nastaví. Poté je zjištěna výška

řetězce `\titleheight` pro zvolený výchozí stupeň písma titulku a je uložena do délkového registru `\titleheight`. Poté je titulek vysazen ve formátu definovaném povinným parametrem příkazu `\tabletitle` a s již zmíněným strutem o výšce zmíněného řetězce umístěném na začátku a na konci titulku.

Nakonec je ještě posunuta souřadnice Y (uchovávána v délkovém registru `\coordinateY`) posunuta dolů o zmíněnou výšku a také o délku uloženou v `\titlespace`.

6.10 Poznámky pod tabulkou

Pro práci s poznámkami pod tabulkou jsou definovány tři různé příkazy. Uvádějí se ve volitelném parametru některého z příkazů pro definici buňky. Všechny tyto příkazy zajišťují sazbu značky poznámky v buňce. První dva z těchto příkazů vytvářejí také samotnou poznámku pod tabulkou:

- příkaz `\tablenote` se dvěma povinnými parametry pro definici poznámky pod tabulkou s vlastní značkou. Značka je dána prvním parametrem, poznámka je dána druhým parametrem,
- příkaz `\tablenotenum` s jedním povinným parametrem pro definici poznámky s automaticky číslovanou značkou, poznámka je dána obsahem parametru,
- příkaz `\tablemark` s jedním povinným parametrem pro uvedení značky poznámky, která je dána obsahem parametru.

Všechny uvedené příkazy je možné používat ve stejné tabulce současně. To je ukázáno na následujícím příkladu, ve kterém je použita již uvedená tabulka, tentokrát s použitím poznámek pod tabulkou (již bez titulku). Definice buněk je následující:

```
\cell[\tabnote{*}]{Buňka v okrajovém sloupci}{A}
\cell[\tabnotenum]{Buňka v prostředním sloupci}{B}
\cell[\tablemark{*}]{C}
```

Výsledná tabulka bude vypadat takto:

Tab. 11: Příklad jednoduché tabulky s poznámkami pod tabulkou vytvořené v prostředí `tableenv`.

A *	B ¹	C *
-----	----------------	-----

* Buňka v okrajovém sloupci

¹ Buňka v prostředním sloupci

Poznámky pod čarou a jejich tvorba v typografickém systému $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ jsou popsány v (Rybička, 2003). Některé rysy těchto poznámek pod čarou byly do jisté

míry převzaty a použity pro poznámky pod tabulkou v tomto stylu. Implementované poznámky pod tabulkou ale s poznámkami pod čarou nijak nekolidují a je možné používat oba typy poznámek v dokumentu současně. Není však např. žádným způsobem ošetřena možnost, že se na stejné stránce objeví poznámka pod čarou i poznámka pod tabulkou se stejnou značkou.

Sazba značek poznámek v tabulce je ještě zmíněna dále u popisu sazby obsahu buňky. Ve stylu jsou pro příkazy pro poznámky pod tabulkou používány tři různé čítače, použití kteréhokoliv z uvedených příkazů inkrementuje jeden nebo více z nich. Jejich přehled je uveden v tab. 12 spolu s tím, které čítače se v případě použití jednotlivých příkazů inkrementují.

Tab. 12: Přehled inkrementace čítačů při použití jednotlivých příkazů.

inkrementovaný čítač	<code>\tablenote</code>	<code>\tablenotenum</code>	<code>\tablemark</code>
<code>\tablenotecount</code>	ano	ano	ne
<code>\tablenumcount</code>	ne	ano	ne
<code>\cellmarkcount</code>	ano	ano	ano

Je také možné v jedné buňce používat více různých uvedených příkazů.

6.10.1 Příkaz `\tablenote`

Tímto příkazem lze vytvořit poznámku se stejnou značkou jako příkazem pro tvorbu automaticky číslované poznámky. Při současném používání těchto dvou příkazů ve stejné tabulce je na to nutné dávat pozor.

Po inkrementaci příslušných čítačů je značka obsažena v prvním parametru příkazu přidána do struktury určené pro uchovávání značek poznámek pod tabulkou (`\marks`) a také do struktury určené pro uchovávání značek poznámek pod tabulkou pro uvedení v buňce (`\cellmarks`). Poznámka je uložena do struktury pro uchovávání poznámek pod tabulkou (`\notes`).

6.10.2 Příkaz `\tablenotenum`

Po inkrementaci všech tří zmíněných čítačů následuje stejná činnost jako u předchozího příkladu s tím rozdílem, že značka poznámky je dána hodnotou již inkrementovaného čítače `\tablenumcount` a poznámku určuje jediný povinný parametr příkazu.

6.10.3 Příkaz `\tablemark`

Je třeba dávat si pozor na to, aby tento příkaz nebyl použit pro sazbu značky, která pro žádnou poznámku není použita. Tato situace není kontrolována.

Pokud je třeba v buňce uvést více značek, je možné tento příkaz (stejně jako oba předchozí) použít ve volitelném parametru buňky vícekrát.

Tento příkaz nejdříve inkrementuje pouze čítač pro počet značek poznámek a poté přidá značku do struktury určené pro uchovávání značek poznámek pod tabulkou pro uvedení v buňce (`\cellmarks`).

6.10.4 Sazba poznámek pod tabulkou, její formátování a implementace

Sazba poznámek pod tabulkou se samozřejmě provádí pouze v případě, že byla alespoň jedna poznámka definována, tedy v případě, že byl v tabulce alespoň jednou použit příkaz `\tablenote` nebo `\tablenotenum`. Stejně jako sazba titulku je také sazba poznámek prováděna prostřednictvím parboxu o šířce tabulky. Při sazbě poznámek pod tabulkou se počítá s jednoznakovými značkami a také se předpokládá, že značka, následná mezera a text poznámky nepřekročí délku jednoho řádku (o délce šířky tabulky). Poznámky pod tabulkou jsou zde uvedeny v pořadí, ve kterém byly po řádcích tabulky definovány. Mezi vlastní tabulkou a poznámky pod tabulkou je možné umístit oddělovací horizontální linku (tato volba je podle výchozího nastavení zapnuta). Při sazbě poznámek pod tabulkou je použit již zmíněný cyklus `whiledo`, který je popsán v (Carlisle, 2001).

Může se ovšem stát, že na poznámky pod tabulkou již nebude na stránce dostatek místa a parbox se umístí např. částečně také mimo rozsah stránky. Na to je třeba si dávat pozor.

Také poznámky pod tabulkou lze formátovat. Kromě již zmíněné možnosti změny stupně písma poznámek jsou definovány následující příkazy pro změnu formátování poznámek pod tabulkou (všechny příkazy slouží pro výchozí nastavení a mají jeden povinný parametr určující novou hodnotu daného nastavení):

- `\noteline` pro zapnutí nebo vypnutí sazby oddělovací linky, výchozí hodnota je 1 (zapnuto),
- `\notelinewidth` pro změnu šířky oddělovací linky, výchozí hodnota je `\thin`,
- `\notelinetopspace` pro změnu šířky vertikální mezery mezi vlastní tabulkou a oddělovací linkou, výchozí hodnota je 5pt,
- `\notelinebottomspace` pro změnu šířky vertikální mezery mezi oddělovací linkou a poznámkami, výchozí hodnota je 5pt,
- `\marknotespace` pro změnu šířky horizontální mezery mezi značkou a poznámkou, výchozí hodnota je 10pt.

Např. nastavení šířky vertikální mezery mezi vlastní tabulkou a oddělovací linkou na 10 pt může být provedeno tímto příkazem:

```
\notelinetopspace{10pt}
```

Důležité je, že i když je vypnuto vykreslování oddělovací linky, stále se vkládají obě vertikální mezery, takže výsledná vertikální mezera mezi vlastní tabulkou a poznámkami pod tabulkou bude rovna součtu těchto dvou šířek.

Nejprve se posune pozice souřadnice Y dolů o šířku dané mezery (mezery mezi vlastní tabulkou a oddělovací linkou) a polovinu její nastavené šířky. Zde se vykreslí oddělovací linka o šířce tabulky (pokud se má tato linka vykreslit). Poté je souřadnice Y posunuta dolů znovu o šířku dané mezery (tentokrát mezery mezi oddělovací

linkou a poznámkami pod tabulkou) a opět polovinu nastavené šířky oddělovací linky. Na tomto místě se vytvoří parbox, do kterého se umístí všechny definované značky a příslušné poznámky pod tabulkou. Mezi značkou a poznámkou je umístěna již zmíněná horizontální meze. To vše se ovšem provádí pouze v případě, že je alespoň jedna poznámka pod tabulkou v dané tabulce definována.

6.11 Buňka

Příkaz `\cell` s jedním povinným a jedním nepovinným parametrem je nejobsáhlejší příkaz, který je ve stylu definován. Zajišťuje kompletní sazbu každé definované buňky, do které v tomto stylu patří také sazba linek okolo buňky. Tento příkaz slouží pro sazbu nesloučené buňky. Ve stylu je však implementována také sazba buňky sloučené přes více sloupců (ne však přes více řádků). Souhrn činností tohoto příkazu je uveden dále v této kapitole, nejdříve jsou však popsány některé části sazby buňky jednotlivě včetně některých činností nutných pro sazbu.

6.11.1 Sloučená buňka

Pro sazbu sloučené buňky slouží příkaz `\merge`. Jeho definice je velmi jednoduchá:

```
\newcommand{\merge}[3][\]{%
\merged=#2
\cell[#1]{#3}
}
```

Důvodem takto jednoduché definice je to, že příkaz `\cell` je implementován tak, že sazba sloučených buněk přes více sloupců v něm již je zahrnuta a jak je vidět v uvedené definici, příkaz pro sazbu sloučené buňky je pomocí něj definovaný. Příkaz `\merge` má oproti příkazu `\cell` jeden povinný parametr navíc (první z povinných parametrů), který určuje počet sloučených sloupců. Volitelný parametr má totožnou funkci jako volitelný parametr příkazu `\cell`. Totéž platí pro druhý povinný parametr a povinný parametr příkazu `\cell`. Sloučená buňka přebírá výchozí nastavení horizontálního zarovnání podle sloupce, ve kterém je příkaz pro sazbu sloučené buňky použit. Počet sloučených sloupců je při použití příkazu `\cell` roven 1 a použití příkazu `\merge` tuto hodnotu modifikuje. Následující dvě definice buňky jsou ekvivalentní:

```
\cell{A}
\merge{1}{A}
```

Buňky budou vypadat následovně:

Tab. 13: Příklad dvou stejných buněk definovaných různými příkazy.

A	A
---	---

6.11.2 Zjištění výšky řádku

Pokud je výška daného řádku, ve které se buňka nachází, zadána (tzn., že do struktury pro uchovávání výšky řádků byla pro daný řádek vložena jiná délka než 0), bude výška řádku tato délka.

V opačném případě je nutné výšku řádku vypočítat. To se provádí způsobem, který již byl zmíněn u možnosti nastavení velikosti písma v různých prvcích tabulky. K této délce se připočítávají délky `\@spacetop`, která určuje velikost mezery mezi horním okrajem buňky a jejím obsahem a `\@spacebottom`, která určuje velikost mezery mezi dolním okrajem buňky a jejím obsahem.

V obou případech se tento rozměr nastaví pomocí příkazu `\rowheight` a je při sazbě buňky často používán.

6.11.3 Zjištění šířky buňky

Šířka sloupce, ve kterém se buňka nachází, se, jak už bylo zmíněno, nastavuje během jeho definice. Tato šířka však platí jen pro nesloučenou buňku, v opačném případě je pro výpočet šířky buňky nutné sečíst šířky všech sloupců, ve kterých buňka leží včetně všech mezilehlých sloupcových mezer. Při tomto výpočtu se používá již zmíněný cyklus `whiledo`.

Rozměr šířky buňky se nastaví pomocí příkazu `\cellwidth` a je, stejně jako výška řádku, při sazbě buňky často používán.

6.11.4 Obsah buňky a mezery od okrajů buňky

`Parbox`, do kterého se umísťuje obsah buňky (daný povinným parametrem příkazu pro sazbu nesloučené buňky nebo druhým povinným parametrem příkazu pro sloučenou buňku), není umístěn přes celý rozměr buňky, ale je odsazen od levého horního rohu buňky dolů o již zmíněnou délku `\@spacetop`, doprava o délku `\@spaceleft`, jeho horizontální rozměr je roven šířce `parboxu` zmenšené o délky `\@spaceleft` a `\@spaceright` a jeho vertikální rozměr je roven výšce `parboxu` zmenšené o délky `\@spacetop` a `\@spacebottom`.

Pro horizontální rozměr tohoto `parboxu` je definován délkový registr `\cellcontentwidth`, pro vertikální rozměr tohoto `parboxu` délkový registr `\cellcontentheight`. Oba jsou nastaveny před sazbu buňky výše popsáním způsobem a při sazbě obsahu buňky jsou použity.

Byly zmíněny délky, které jsou použity pro nastavení obsahu buňky od jejích okrajů. Lze je měnit pomocí příkazů uvedených v tab. 14. Jsou definovány příkazy pro změnu jedné mezery i více mezer najednou (na stejnou velikost) a to pro změnu

výchozího i lokálního nastavení. Všechny tyto příkazy mají jeden povinný parametr určující novou hodnotu dané mezery. Ve výchozím nastavení jsou všechny čtyři mezery nastaveny na 5pt.

Tab. 14: Seznam příkazů pro změnu nastavení mezer mezi obsahem buňky a okrajem buňky.

mezera	lokální změna	výchozí změna
horní mezera	<code>\spacetop</code>	<code>\defspacetop</code>
levá mezera	<code>\spaceleft</code>	<code>\defspaceleft</code>
dolní mezera	<code>\spacebottom</code>	<code>\defspacebottom</code>
pravá mezera	<code>\spaceright</code>	<code>\defspaceright</code>
levá a pravá mezera	<code>\spaceh</code>	<code>\defspaceh</code>
horní a dolní mezera	<code>\spacev</code>	<code>\defspacev</code>
všechny mezery	<code>\spacehv</code>	<code>\defspacehv</code>

Pokud je v některé buňce nutné změnit její výchozí horizontální zarovnání obsahu, lze k tomu použít příkaz `\alignh` s jedním povinným parametrem, který určuje nové zarovnání. V parametru se používají stejné znaky jako při definici sloupce (převzaty z prostředí `tabular`).

Je možné nastavit také vertikální zarovnání obsahu buňky. Slouží k tomu následující příkazy s jedním povinným parametrem určujícím pozici:

- `\alignv` pro lokální změnu,
- `\defalignv` pro nastavení výchozího vertikálního zarovnání.

Pro vertikální pozici se používá přímo jeden z volitelných parametrů příkazu `parbox`. Volitelné parametry `parboxu` a možná vertikální zarovnání jsou popsány v (Rybička, 2003). Možné hodnoty parametru jsou následující:

- `t` pro zarovnání nahoru,
- `c` pro zarovnání na střed,
- `b` pro zarovnání dolů.

Výchozí hodnotou je `c`, tedy zarovnání na střed.

6.11.5 Linky okolo buňky

Sazba linek okolo buňky se ve stylu řídí následujícími pravidly:

- do mezerádkových a mezisloupcových mezer jsou podle výchozího nastavení umísťovány linky o šířce těchto mezer,
- každá buňka zajišťuje sazbu dolní a pravé linky,
- pouze buňky v prvním řádku zajišťují sazbu také horní linky,
- pouze buňky v prvním sloupci zajišťují sazbu také levé linky,
- podle výchozího nastavení každá horizontální linka vede od poloviny předchozí mezisloupcové mezery do poloviny následující mezisloupcové mezery s výjimkou okrajových sloupců, kde je vždy na straně u okraje tabulky linka dotažena přes celou okrajovou mezisloupcovou mezeru,

- podle výchozího nastavení každá vertikální linka vede od poloviny předchozí meziřádkové mezery do poloviny následující meziřádkové mezery.

Jediná buňka, která tedy zajišťuje sazbu všech čtyř linek okolo sebe je první buňka (buňka umístěná v levém horním rohu tabulky). Kdyby bylo implementováno zajišťování sazby všech linek okolo buňky každou buňkou, bylo by v případě dvou sousedících buněk možné formátovat linku mezi nimi prostřednictvím obou buněk, což by mohlo způsobovat problémy. V současné implementaci lze každou linku v tabulce lokálně formátovat prostřednictvím pouze jediné buňky, což zmíněným problémům předchází.

Na obr. 9 je vidět, jaké situaci předchází nastavení rozdílného dotahu horizontálních linek na straně u okrajů tabulky (v tomto případě je rozhodující změna levého dotahu dolní horizontální linky první buňky v poslední řádce).



Obr. 9: Detail levého dolního rohu tabulky bez změny dotahu horizontální linky (vlevo) a s její již uvedenou změnou (vpravo).

Linky okolo buněk lze samozřejmě formátovat. Je implementováno zapínání a vypínání sazby linek a formátování dotahů linek. K dalšímu formátování linek (jedná se o šířku čáry, barvu čáry a styl čáry) se přímo využívá formátování čar popsané v (Tantau aj., 2010).

Všechny definované příkazy pro formátování linek okolo buňky mají jeden povinný parametr udávající hodnotu daného parametru linky.

Zapnutí nebo vypnutí sazby každé linky lze provést pomocí příkazů uvedených v tab. 15. V povinném parametru se uvádí 1 pro zapnutí sazby a 0 pro její vypnutí. Všechny čtyři linky jsou podle výchozího nastavení zapnuty.

Tab. 15: Seznam příkazů pro zapnutí a vypnutí sazby linek okolo buňky.

linka	lokální změna	výchozí změna
horní linka	<code>\linetop</code>	<code>\deflinetop</code>
levá linka	<code>\lineleft</code>	<code>\deflineleft</code>
dolní linka	<code>\linebottom</code>	<code>\deflinebottom</code>
pravá linka	<code>\lineright</code>	<code>\deflineright</code>
horizontální linky	<code>\linesh</code>	<code>\deflinesh</code>
vertikální linky	<code>\linesv</code>	<code>\deflinesv</code>
všechny linky	<code>\lineshv</code>	<code>\deflineshv</code>

Dotahy linek se udávají jako část mezisloupcové nebo meziřádkové mezery, do které má linka zasahovat. Výchozí nastavení dotahů je polovina všech těchto mezer,

tedy hodnota 0.5. Jak již ale bylo zmíněno, neplatí to pro dotahy horizontálních linek na okrajích tabulky. V prvním sloupci je dotah těchto linek směrem doleva zvýšen vždy o 0.5, stejně tak horizontálních linek v posledním sloupci směrem doprava. Podle výchozího nastavení je tedy celkový zmíněný dotah roven hodnotě 1, což znamená dotah přes celou mezisloupcovou mezeru. Příkazy pro nastavení dotahů linek jsou uvedeny v tab. 16.

Tab. 16: Seznam příkazů pro nastavení dotahů linek okolo buňky.

dotah	lokální změna	výchozí změna
levý dotah horní linky	<code>\linespacetopleft</code>	<code>\deflinespacetopleft</code>
pravý dotah horní linky	<code>\linespacetopright</code>	<code>\deflinespacetopright</code>
horní dotah levé linky	<code>\linespacelefttop</code>	<code>\deflinespacelefttop</code>
dolní dotah levé linky	<code>\linespaceleftbottom</code>	<code>\deflinespaceleftbottom</code>
levý dotah dolní linky	<code>\linespacebottomleft</code>	<code>\deflinespacebottomleft</code>
pravý dotah dolní linky	<code>\linespacebottomright</code>	<code>\deflinespacebottomright</code>
horní dotah pravé linky	<code>\linespacerighttop</code>	<code>\deflinespacerighttop</code>
dolní dotah pravé linky	<code>\linespacerightbottom</code>	<code>\deflinespacerightbottom</code>
nastavení obou dotahů horní linky	<code>\linespacet</code>	<code>\deflinespacet</code>
nastavení obou dotahů levé linky	<code>\linespacel</code>	<code>\deflinespacel</code>
nastavení obou dotahů dolní linky	<code>\linespaceb</code>	<code>\deflinespaceb</code>
nastavení obou dotahů pravé linky	<code>\linespacer</code>	<code>\deflinespacer</code>
nastavení všech dotahů horizontálních linek	<code>\linespaceh</code>	<code>\deflinespaceh</code>
nastavení všech dotahů vertikálních linek	<code>\linespacev</code>	<code>\deflinespacev</code>
nastavení všech dotahů všech linek	<code>\linespacehv</code>	<code>\deflinespacehv</code>

Lze použít také záporné hodnoty (v tom případě linka nedosáhne k mezisloupcové (příp. meziřádkové) mezeře, ale také lze použít hodnoty vyšší než 1 (v tom případě linka přesáhne mezisloupcovou (nebo meziřádkovou) mezeru. Dotahy samozřejmě nemají žádný účinek v případě nulových mezisloupcových (příp. meziřádkových) mezer.

Dotahy uvnitř tabulky nefungují úplně optimálně. Pravděpodobně je to dáno tím, že v bodě konce některé linky velmi často začíná linka další.

Šířka linek se podle výchozího nastavení určuje podle šířky mezisloupcové nebo meziřádkové mezery, ve které se linka nachází (tomu odpovídá nastavení šířky na 0). Šířku kterékoliv linky je samozřejmě možné změnit. Příkazy, které k tomu slouží, jsou uvedeny v tab. 17.

Tab. 17: Seznam příkazů pro nastavení šířky linek.

linka	lokální změna	výchozí změna
horní linka	<code>\linetopwidth</code>	<code>\deflinetopwidth</code>
levá linka	<code>\lineleftwidth</code>	<code>\deflineleftwidth</code>
dolní linka	<code>\linebottomwidth</code>	<code>\deflinebottomwidth</code>
pravá linka	<code>\linerightwidth</code>	<code>\deflinerightwidth</code>
horizontální linky	<code>\lineshwidth</code>	<code>\deflineshwidth</code>
vertikální linky	<code>\linesvwidth</code>	<code>\deflinesvwidth</code>
všechny linky	<code>\lineshwidth</code>	<code>\deflineshwidth</code>

Je důležité si uvědomit, že při změně šířky linky se nezmění šířka mezery, ve které je linka umístěna. Při použití širší linky než šířka této mezery, bude linka rovnoměrně zasahovat do prostoru buněk, mezi kterými je umístěna a tím ho v podstatě zmenšovat a při použití užší linky bude prostor v buňkách naopak v podstatě zvětšovat.

Specifikace barvy je popsána v (Tantau aj., 2010). Pro nastavení barvy linek jsou definovány příkazy uvedené v tab. 18. Obsahem povinného parametru je barva linky, podle výchozího nastavení je barva všech linek černá, tedy nastavena na `black`.

Tab. 18: Seznam příkazů pro nastavení barvy linek.

linka	lokální změna	výchozí změna
horní linka	<code>\linetopcolor</code>	<code>\deflinetopcolor</code>
levá linka	<code>\lineleftcolor</code>	<code>\deflineleftcolor</code>
dolní linka	<code>\linebottomcolor</code>	<code>\deflinebottomcolor</code>
pravá linka	<code>\linerightcolor</code>	<code>\deflinerightcolor</code>
horizontální linky	<code>\lineshcolor</code>	<code>\deflineshcolor</code>
vertikální linky	<code>\linesvcolor</code>	<code>\deflinesvcolor</code>
všechny linky	<code>\lineshcolor</code>	<code>\deflineshcolor</code>

Práce se styly linek je popsána v (Tantau aj., 2010). Jak již bylo zmíněno, v současné implementaci jsou zahrnuty tři různé styly linky, které se zadávají číslicí. Jsou to:

- 1 pro plnou linku
- 2 pro tečkovanou linku
- 3 pro čárkovanou linku

Příkazy pro jejich nastavení jsou uvedeny v tab. 19. Výchozí nastavení stylu všech linek je 1 (plná čára).

Tab. 19: Seznam příkazů pro nastavení stylu linek.

linka	lokální změna	výchozí změna
horní linka	<code>\linetopstyle</code>	<code>\deflinetopstyle</code>
levá linka	<code>\lineleftstyle</code>	<code>\deflineleftstyle</code>
dolní linka	<code>\linebottomstyle</code>	<code>\deflinebottomstyle</code>
pravá linka	<code>\linerightstyle</code>	<code>\deflinerightstyle</code>
horizontální linky	<code>\lineshstyle</code>	<code>\deflineshstyle</code>
vertikální linky	<code>\linesvstyle</code>	<code>\deflinesvstyle</code>
všechny linky	<code>\lineshvstyle</code>	<code>\deflineshvstyle</code>

6.11.6 Pozadí buňky

Barva pozadí buňky je řešena pomocí sazby barevného vyplněného obdélníku pokrývající celou buňku a polovinu obou mezirádkových i obou mezisloupcových mezer okolo buňky. Sazba obdélníku je popsána v (Tantau aj., 2010). Obsah povinného parametru příkazů, které pro nastavení barvy pozadí slouží, definuje barvu pozadí. Tyto příkazy jsou definovány dva:

- `\bcolor` pro lokální nastavení barvy pozadí,
- `\defbcolor` pro výchozí nastavení barvy pozadí.

Nastavení na hodnotu 0 sazbu pozadí buňky vypíná, což je také výchozí nastavení pozadí buněk.

6.11.7 Rámeček buňky

Rámeček buňky je řešen jako obdélník, sazba obdélníku je popsána v (Tantau aj., 2010), který je vzdálený od všech okrajů buňky stejně. Pro rámeček buňky jsou definovány následující příkazy:

- `\frame` pro změnu lokálního nastavení sazby rámečku,
- `\defframe` pro změnu výchozího nastavení sazby rámečku,
- `\framewidth` pro změnu lokálního nastavení šířky rámečku,
- `\defframewidth` pro změnu výchozího nastavení šířky rámečku,
- `\framecolor` pro změnu lokálního nastavení barvy rámečku,
- `\defframecolor` pro změnu výchozího nastavení barvy rámečku,

- `\framespace` pro změnu lokálního nastavení vzdálenosti rámečku od okrajů buňky,
- `\defframespace` pro změnu výchozího nastavení vzdálenosti rámečku od okrajů buňky.

Podle výchozího nastavení je sazba rámečku vypnutá (nastavena na hodnotu 0), jeho šířka je nastavena na hodnotu `\thin`, barva je černá (nastavena na `black`) a jeho vzdálenost od okraje buňky je nastavena na `2.5pt`, ve výchozím nastavení tedy polovinu všech vzdáleností obsahu od okrajů buňky. Vzdáleností rámečku od okrajů buňky se rozumí vzdálenost od středu čar obdélníku tvořícího rámeček.

Jako vzdálenost rámečku od okrajů buňky lze uvést i takovou hodnotu, která v podstatě způsobí zesílení linek okolo buňky. Je možné zapsat také hodnotu 0 nebo dokonce hodnotu zápornou, která rámeček (přesněji středy čar) umístí mimo prostor buňky.

6.11.8 Přeskrtnutí buňky

Přeskrtnutí buňky je řešeno jako jedna nebo dvě čáry vedeny z jednoho rohu buňky do rohu protějšího. Pro přeskrtnutí buňky jsou definovány následující příkazy:

- `\stroke` pro změnu lokálního nastavení sazby přeskrtnutí,
- `\defstroke` pro změnu výchozího nastavení sazby přeskrtnutí,
- `\strokewidth` pro změnu lokálního nastavení šířky čar přeskrtnutí,
- `\defstrokewidth` pro změnu výchozího nastavení šířky čar přeskrtnutí,
- `\strokecolor` pro změnu lokálního nastavení barvy čar přeskrtnutí,
- `\defstrokecolor` pro změnu výchozího nastavení barvy čar přeskrtnutí.

Styly přeskrtnutí buňky jsou následující:

- 1 pro jednu čáru vedenou z levého dolního rohu do pravého horního rohu,
- 2 pro jednu čáru vedenou z levého horního rohu do pravého dolního rohu,
- 3 pro obě výše zmíněné čáry.

Podle výchozího nastavení je sazba přeskrtnutí buňky vypnuta (nastavena na hodnotu 0), šířka čar je nastavena na `\thin` a barva čar je nastavena na černou (`black`).

6.11.9 Shrnutí sazby buněk

Příkaz nejprve provede zjištění výchozích hodnot parametrů, kterých je v případě tohoto příkazu velké množství a některé hodnoty je třeba vypočítat. Provádí se také nastavení některých čítačů a definice struktury pro uchovávání značek poznámek pod tabulkou v buňce.

Před sazbou obsahu buňky je nutné provést výpočet výšky řádku textu v buňce (to již bylo zmíněno, provádí se podle výšky a hloubky řetězce (Ěj)). Před i za obsah buňky se vkládá již zmíněný strut o zjištěné výšce. Za obsahem buňky je ještě případně provedena sazba všech značek poznámek pod tabulkou, které byly ve volitelném parametru definovány. K tomu se využívá již zmíněný cyklus `whiledo`.

Následně proběhne sazba všech linek okolo buňky v pořadí horní linka, levá linka, dolní linka a pravá linka a následně jsou vysázeny zbývající prvky buňky, tedy pozadí buňky, rámeček buňky a přeškrtnutí buňky. Samozřejmě se provede sazba jen těch prvků, jejichž sazba je zapnutá.

Následně se ještě souřadnice `X` nastaví na horní levý roh následující buňky (i v případě, že již další buňka na řádku není, protože v takovém případě příkaz pro ukončení řádku `\rowend` nastaví souřadnice na začátek dalšího řádku), inkrementuje čítač pořadí sloupce `\column` a nastaví čítač počtu sloučených buněk na 1.

6.12 Tvorba pořadové sazby, různých typů tabulek a diagramů v prostředí `\tableenv`

Pro názornost je níže uvedeno, jaké nastavení ve stylu provést pro tvorbu pořadové sazby a různých typů tabulek. Také je uvedeno, jak postupovat při tvorbě diagramů. Pořadová sazba, tabulky a jejich typy a diagramy jsou popsány v (Pop, Flégr, Pop, 1989).

6.12.1 Tvorba pořadové sazby

Pro tvorbu pořadové sazby je nutné vypnout všechny čtyři linky okolo buňky v celé tabulce:

Tab. 20: Příklad pořadové sazby vytvořené v prostředí `\tableenv`.

Seznam účastníků poháru žáků v běhu

Číslo	Jméno	Trať	Rok narození	Město
1	Bárta Miroslav	100 m	2001	Břeclav
2	Bylný Milan	100 m, 200 m	2000	Brno
3	Kadlec Jaromír	200 m	2000	Vyškov
4	Krusta Ondřej	200 m	2000	Brno
5	Novák Jan	100 m	1999	Brno
6	Novák Tomáš	100 m	1999	Kuřim
7	Pavelka Petr	100 m, 200 m	2001	Brno
8	Sup Václav	100 m, 200 m	2000	Tišnov
9	Štrbák David	200 m	2001	Brno
10	Žák Michal	100 m, 200 m	2000	Blansko

Tato pořadová sazba byla získána následujícím zápisem:

```

\deflineshv{0}

\begin{tableenv}
\colspace\col[1.5cm]{1}\colspace\col[3.1cm]{1}\colspace
\col[2.8cm]{1}\colspace\col[3.1cm]{1}\colspace\col[1.8cm]{1}\colspace

\rowend
\merge{5}{\textbf{Seznam účastníků poháru žáků v běhu}}\rowend
\cell{\textbf{Číslo}}\cell{\textbf{Jméno}}\cell{\textbf{Trať}}
\cell{\textbf{Rok narození}}\cell{\textbf{Město}}\rowend
\cell{1}\cell{Bárta Miroslav}\cell{100 m}\cell{2001}\cell{Břeclav}
\rowend
\cell{2}\cell{Bylný Milan}\cell{100 m, 200 m}\cell{2000}\cell{Brno}
\rowend
\cell{3}\cell{Kadlec Jaromír}\cell{200 m}\cell{2000}\cell{Vyškov}
\rowend
\cell{4}\cell{Krusta Ondřej}\cell{200 m}\cell{2000}\cell{Brno}\rowend
\cell{5}\cell{Novák Jan}\cell{100 m}\cell{1999}\cell{Brno}\rowend
\cell{6}\cell{Novák Tomáš}\cell{100 m}\cell{1999}\cell{Kuřim}\rowend
\cell{7}\cell{Pavelka Petr}\cell{100 m, 200 m}\cell{2001}\cell{Brno}
\rowend
\cell{8}\cell{Sup Václav}\cell{100 m, 200 m}\cell{2000}\cell{Tišnov}
\rowend
\cell{9}\cell{Štrbák David}\cell{200 m}\cell{2001}\cell{Brno}\rowend
\cell{10}\cell{Žák Michal}\cell{100 m, 200 m}\cell{2000}
\cell{Blansko}\rowend
\end{tableenv}

\deflineshv{1}

```

6.12.2 Tvorba tabulek

Pořadová sazba v tabulkové úpravě, tabulka otevřená a tabulka uzavřená jsou popsány v (Pop, Flégr, Pop, 1989). Všechny tyto typy sazby lze v prostředí `\tableenv` vytvářet pomocí různého nastavení sazby linek okolo buněk v tabulce. Následující tabulka byla vytvořena v prostředí `\tableenv` a využívá několik formátování zmíněných v této kapitole:

Tab. 21: Příklad tabulky vytvořené v prostředí `\tableenv`.

Seznam účastníků poháru žáků v běhu				
Číslo	Jméno	Trat' *	Rok narození	Město
1	Bárta Miroslav	100 m	2001	Břeclav
2	Bylný Milan	100 m, 200 m	2000	Brno
3	Kadlec Jaromír	200 m	2000	Vyškov
4	Krusta Ondřej	200 m	2000	Brno
5	Novák Jan	100 m	1999	Brno
6	Novák Tomáš	100 m	1999	Kuřim
7	Pavelka Petr	100 m, 200 m	2001	Brno
8	Sup Václav	100 m, 200 m	2000	Tišnov
9	Štrbák David	200 m	2001	Brno
10	Žák Michal	100 m, 200 m	2000	Blansko

* Každý účastník se může účastnit běhu na jedné nebo obou tratích.

Tato tabulka byla získána následujícím zápisem:

```

\begin{tableenv}
\colspace\col[1.5cm]{r}\colspace\col[3.1cm]{l}\colspace
\col[2.8cm]{c}\colspace\col[3.1cm]{c}\colspace\col[1.8cm]{l}\colspace

\rowend
\defbcolor{yellow}\merge[\alignh{c}]{5}
{\textbf{Seznam účastníků poháru žáků v běhu}} \rowend
\cell{\textbf{Číslo}}\cell{\textbf{Jméno}}\cell[\tablenote{*}]{Každý
účastník se může účastnit běhu na jedné nebo obou tratích.}
{\textbf{Trat'}}\cell{\textbf{Rok narození}}\cell{\textbf{Město}}\rowend
\defbcolor{0}\cell{1}\cell{Bárta Miroslav}\cell{100 m}\cell{2001}
\cell{Břeclav}\rowend
\defbcolor{gray!50}\cell{2}\cell{Bylný Milan}\cell{100 m, 200 m}
\cell{2000}\cell{Brno}\rowend
\defbcolor{0}\cell{3}\cell{Kadlec Jaromír}\cell{200 m}\cell{2000}
\cell{Vyškov}\rowend
\defbcolor{gray!50}\cell{4}\cell{Krusta Ondřej}\cell{200 m}
\cell{2000}\cell{Brno}\rowend

```

```

\defbcolor{0}\cell{5}\cell{Novák Jan}\cell{100 m}\cell{1999}
\cell{Brno}\rowend
\defbcolor{gray!50}\cell{6}\cell{Novák Tomáš}\cell{100 m}\cell{1999}
\cell{Kuřim}\rowend
\defbcolor{0}\cell{7}\cell{Pavelka Petr}\cell{100 m, 200 m}
\cell{2001}\cell{Brno}\rowend
\defbcolor{gray!50}\cell{8}\cell{Sup Václav}\cell{100 m, 200 m}
\cell{2000}\cell{Tišnov}\rowend
\defbcolor{0}\cell{9}\cell{Štrbák David}\cell{200 m}\cell{2001}
\cell{Brno}\rowend
\defbcolor{gray!50}\cell{10}\cell{Žák Michal}\cell{100 m, 200 m}
\cell{2000}\cell{Blansko}\rowend
\end{tableenv}

\defbcolor{0}

```

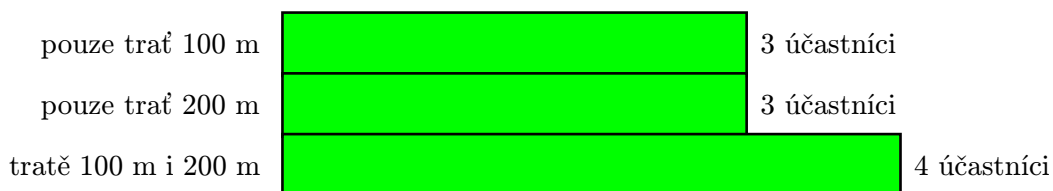
6.12.3 Tvorba diagramů

Jedním z méně důležitých požadavků na styl byla možnost použití stylu pro možnost sazby diagramů. Jednoduchou sazbu jednoho typu diagramu současná implementace umožňuje a sice sloupcového diagramu na šířku, popsáno v (Pop, Flégr, Pop, 1989).

Postup pro sazbu zmíněného typu diagramu je následující:

První sloupec může sloužit k popisu jednotlivých sloupců. Dále je třeba nadefinovat takový počet stejně širokých sloupců, aby bylo možné tímto počtem celočíselně vydělit (může se také použít zaokrouhlování, pokud nebude vadit nižší přesnost) hodnoty ve všech sloupcích. Délky sloupců se poté definují jako buňka sloučená od druhého sloupce přes požadovaný počet sloupců. Může být vhodné také sloučit zbývající buňky na řádku, protože pokud má být za sloupkem uvedena hodnota, šířka jednoho sloupce nemusí dostačovat. Pro uvedení hodnoty u dlouhých sloupců také může být nutné vytvořit na konci řádků širší sloupec, aby bylo pro uvedení hodnoty vždy dostatek místa. Pro ohraničení sloupců je vhodné užívat rámeček buňky místo linek (jejich sazbu je vhodné naopak vypnout). Podle toho, jak má diagram vypadat, může být potřeba použít další nastavení. Některá jsou patrná z příkladu diagramu vytvořeného v prostředí `\tableenv` na obr. 10.

Počty účastníků poháru žáků v běhu podle tratí



Obr. 10: Příklad diagramu vytvořeného pomocí prostředí `tableenv`.

Tento diagram byl získán následujícím zápisem:

```
\deflineshv{0}
\defframespace{-\thin}

\begin{tableenv}
\colspace\col[3.8cm]{r}\colspace\col{1}\colspace\col{1}\colspace
\col{1}\colspace\col{1}\colspace\col[2.3cm]{1}\colspace

\rowend
\merge{4}{\textbf{Počty účastníků poháru žáků v běhu podle tratí}}
\rowend
\cell{pouze trať 100 m}\merge[\bcolor{green}\frame{1}]{3}{}
\merge{2}{3 účastníci}\rowend
\cell{pouze trať 200 m}\merge[\bcolor{green}\frame{1}]{3}{}
\merge{2}{3 účastníci}\rowend
\cell{tratě 100 m i 200 m}\merge[\bcolor{green}\frame{1}]{4}{}
\cell{4 účastníci}\rowend
\end{tableenv}
```

Styl sice umožňuje také např. sazbu sloupcového diagramu na výšku, který je rovněž zmíněn v (Pop, Flégr, Pop, 1989), ale jeho sazba je v tomto stylu složitá. Styl by bylo možné pro sazbu diagramů více přizpůsobit, ale hlavním použitím stylu je sazba tabulek.

6.12.4 Nasazení stylu v projektu **T_EXonWeb**

Projekt T_EXonWeb bude v blízké době aktualizován a tento styl v něm není prozatím k dispozici. Styl však zde již byl otestován a po aktualizaci projektu se zde počítá s umístěním tohoto stylu a bude tak možné si jej zde vyzkoušet a používat.

7 Zhodnocení

Současná implementace stylu zahrnuje následující:

- možnost tvorby různých tabulek a diagramů
 - je umožněna tvorba pořadové sazby a díky širokým možnostem nastavení linek okolo buněk také tvorba různých typů tabulek
 - způsob implementace je také vhodný pro tvorbu sloupcových diagramů na šířku
- formátování tabulky jako celku
 - možnost nastavení vertikální mezery nad tabulkou a pod ní
 - volitelně lze nad tabulkou uvádět její titulek, u kterého lze nastavit jeho horizontální zarovnání vzhledem k šířce tabulky. Automaticky uváděný text titulku je možné předefinovat a tento text může, ale také nemusí, obsahovat číslování tabulky
 - styl umožňuje tvorbu dvou typů poznámek pod tabulkou (s uživatelem definovanou značkou a s automatickým číslováním). Oba typy poznámek je možné používat současně ve stejné tabulce a také je definován příkaz pouze pro sazbu značky poznámky
 - lze přednastavit stupeň písma textů v buňkách tabulky, titulku tabulky i poznámek pod tabulkou. Nastavený stupeň písma se mj. používá pro určení výchozí výšky řádku
 - je implementována práce s vrstvami, pomocí kterých lze mj. řešit správné křížení linek
- formátování sloupců a řádků
 - je umožněno nastavení šířky sloupce a dále výchozí šířky sloupce, která se použije, pokud šířka sloupce není určena a také nastavení šířky mezisloupcových mezer
 - je umožněno nastavení výšky řádku a dále výchozí výšky řádku, která je nastavena na hodnotu vypočítanou podle nastaveného stupně písma a horní a dolní mezery mezi obsahem buňky a jejími okraji (toto nastavení předpokládá text o délce jednoho řádku). Lze si samozřejmě zvolit jinou výchozí výšku a také je umožněno nastavení šířky meziřádkových mezer
 - je umožněno nastavit výchozí horizontální zarovnání obsahu buněk ve sloupci
 - pro sloupec lze definovat již zmíněné výchozí horizontální zarovnání a šířku sloupce, pro mezisloupcovou mezeru lze definovat její šířku a styl linky v ní umístěné, pro řádek lze (pomocí příkazu pro ukončení řádku) definovat výšku následujícího řádku a šířku meziřádkové mezery pod následujícím řádkem. Dále lze formátovat řádkově souvislé oblasti, které mohou zahrnovat jeden řádek, jeho část, nebo např. celou tabulku nebo i všechny tabulky v dokumentu. Takto lze měnit téměř veškerá nastavení a formátování buněk a linek okolo nich
- formátování buněk

- pro každou buňku lze zvolit horizontální a vertikální zarovnání jejího obsahu (lze nastavit také výchozí vertikální zarovnání obsahu buněk)
- pro každou buňku lze nastavit vzdálenosti obsahu buňky od okrajů buňky (lze nastavit také výchozí vzdálenosti obsahu od okrajů buněk)
- lze slučovat buňky přes více sloupců
- je možné nastavit barvu pozadí buňky
- je umožněno zapnout sazbu rámečku buňky a nastavit šířku rámečku, vzdálenost od okrajů (lze nastavit také nulovou nebo dokonce zápornou vzdálenost) a také jeho barvu
- je možné nastavit přeškrtnutí buňky, zahrnuty jsou tři styly přeškrtnutí
- formátování linek
 - styl zahrnuje také dvě slovně pojmenované délky, které lze využívat např. pro nastavení šířky linek, ale i pro jiná nastavení
 - lze nastavit šířku linky, její barvu, zahrnuta je sazba linek třemi různými styly čar (plná čára, tečkovaná čára a čárkovaná čára) a také lze nastavit dotah linek

V implementovaném stylu jsou tedy zahrnuty téměř všechny hlavní požadavky a také většina méně důležitých požadavků.

7.1 Omezení stylu

Největší omezení stylu již bylo zmíněno v předchozí kapitole a jedná se o absenci automatického určení šířky buňky podle buňky s nejširším obsahem ve sloupci a automatického určení výšky buňky podle buňky s nejvyšším obsahem v řádku (automatické určení výšky buňky částečně implementováno je, lze ale použít jen v případě, kdy obsah buněk není delší než jeden řádek). Ostatní hlavní požadavky splněny byly. Z méně podstatných požadavků nebyly do implementace zahrnuty tyto:

- možnost vytvářet tabulky s přesahem na více stran
- možnost slučování buněk přes více řádků

Omezení v provedené implementaci již lze považovat za podstatně méně závažná a také byla zmíněna v předchozí kapitole vždy u prvku tabulky, kterého se týká. Jedná se zejména o:

- nutnost využít možnosti nastavit vertikální mezeru mezi titulkem a vlastní tabulkou, pokud je titulek delší než šířka tabulky (důvod je stejný jako u největšího omezení stylu zmíněného výše),
- možnost, že poznámky pod tabulkou budou umístěny např. mimo rozsah stránky, pokud již pro ně není na stránce dostatek místa. V takovém případě je třeba, aby uživatel např. umístil tabulku až na další stránku.

Při případném budoucím rozšíření stylu by bylo vhodné zaměřit se především na zmíněná omezení. Bylo by také vhodné např. rozšířit možnosti formátování sloupců a mezisloupcových mezer.

7.2 Práce se stylem

Se stylem se pracuje relativně jednoduše za předpokladu, že se uživatel seznámí se strukturou používání stylu a bude znát základní ve stylu definované příkazy. Používání pokročilejšího formátování vyžaduje hlubší znalost příkazů, aby práce probíhala efektivně. Pokud uživateli nevyhovuje výchozí nastavení, lze je upravit v souboru `tableenvdefaults.sty`, kde je nastavení rozděleno podle prvků tabulky, ke kterému nastavení patří, příp. podle stejného nastavení pro různé prvky tabulky. Při srovnání stylu např. s prostředím `tabular` může některým uživatelům vadit jistá nižší přehlednost zápisu.

Styl lze použít pro sazbu většiny tabulek s běžně používaným formátováním, i když se zmíněnými omezeními může být tvorba pracnější než např. ve zmíněném prostředí `tabular`.

8 Závěr

Tato práce se zabývala teoretickými poznatky o tabulkách a přístupy k tvorbě tabulek v některých programových nástrojích, (především ale v typografickém systému $\text{\TeX}/\text{\LaTeX}$) a byly uvedeny způsoby, jak se v nich k tvorbě tabulek přistupuje. Praktická část této práce je věnována tvorbě stylu pro sazbu tabulek právě v typografickém systému $\text{\TeX}/\text{\LaTeX}$.

Součástí této práce je analýza třiceti dokumentů obsahujících tabulky, jejíž výsledky následně přispěly k určení formátování tabulek, které se používá nejčastěji. Na základě toho byly definovány požadavky na zmíněný styl. Ty zahrnovaly nejen požadavky vyplývající z předchozí analýzy, ale část požadavků vyplynula ze zmíněných teoretických poznatků a některé požadavky jsou zcela zřejmé.

Dále byly zvoleny nástroje dostupné v typografickém systému $\text{\TeX}/\text{\LaTeX}$, které budou využity pro implementaci stylu. Byl vyžadován nástroj pro rozmísťování textu (za který byly zvoleny parboxy) a dále nástroj pro sazbu veškerých grafických prvků tabulky (za který byly zvoleny balíky PGF a TikZ).

Velká část práce je následně věnována popisu implementace stylu, ve které jsou popsány všechny implementované prvky tabulek, způsob jejich implementace, definované příkazy a jejich používání. V souborech jsou hojně využívány komentáře, které pomáhají k tomu, aby se se stylem lépe seznámil uživatel, který jej chce používat, ale také uživatel, který by chtěl funkcionalitu dále rozšiřovat nebo měnit. Takovému uživateli může samozřejmě k podrobnějšímu seznámení posloužit také popis implementace stylu v této práci.

V současné implementaci byla splněna naprostá většina definovaných požadavků a výsledný styl je použitelný pro relativně snadnou sazbu nejen tabulek, ale také určitého typu diagramu. Vzhledem ale k některým omezením tohoto stylu může být sazba tabulek pracnější než v některém z běžně užívaných prostředí pro sazbu tabulek.

Po odstranění alespoň některých omezení se ale tento styl může stát plnohodnotným nástrojem, který umožňuje poměrně komplexní sazbu tabulek. Styl také samozřejmě může být dále rozšiřován.

Do budoucna se počítá s nasazením tohoto stylu v projektu \TeX onWeb, kde jej bude možné používat.

9 Použitá literatura

- CARLISLE, D. *The ifthen package*. Vytvořeno: 2001-05-26. Verze souboru: v1.1c. Poslední revize: 2001-05-26. Dostupné v instalaci distribuce TeX Live 2012. Z příkazového řádku příkazem `texdoc ifthen`.
- JANOVSKÝ, D. *Jak psát web, návod na html stránky. Formátovací možnosti tabulek v HTML 4* [online]. [cit. 2013-05-14]. Poslední aktualizace: 2012-12-06. ISSN 1801-0458. Dostupné z: <http://www.jakpsatweb.cz/tabulky-nove.html>.
- JANOVSKÝ, D. *Jak psát web, návod na html stránky. Formátování tabulek v HTML* [online]. [cit. 2013-05-14]. Poslední aktualizace: 2012-12-06. ISSN 1801-0458. Dostupné z: <http://www.jakpsatweb.cz/tabulky-format.html>.
- JANOVSKÝ, D. *Jak psát web, návod na html stránky. Tabulky v HTML stránkách* [online]. [cit. 2013-05-14]. Poslední aktualizace: 2012-12-06. ISSN 1801-0458. Dostupné z: <http://www.jakpsatweb.cz/tabulky.html>.
- OLŠÁK, P. *TEXbook naruby* [online]. 2. vydání. Brno: Konvoj, 2001. 468 s. ISBN 80-7302-007-6. Dostupné z: <ftp://math.feld.cvut.cz/pub/olsak/tbn/tbn.pdf>.
- PECINOVSKÝ, J. *Word 2007 podrobný průvodce*. První vydání. Praha: Grada Publishing, a.s., 2007. 228 s. ISBN 978-80-247-1959-7.
- PÍSEK, S. *HTML – tvorba jednoduchých internetových stránek*. 2., aktualizované a doplněné vydání. Praha: Grada Publishing, a.s., 2006. 108 s. ISBN 978-80-247-1767-8.
- POP, P., FLÉGER, J., POP, V. *Sazba I. Ruční sazba pro I. a II. ročník střední průmyslové školy grafické a střední odborné učiliště polygrafické*. 2. vydání. Praha: Státní pedagogické nakladatelství, 1989. 188 s.
- PŘICHYSTAL, J. *TEXonWeb* [online]. [cit. 2013-05-20]. Dostupné z: <http://tex.mendelu.cz/>.
- RYBIČKA, J. *ETEX pro začátečníky*. 3. vydání. Brno: Konvoj, 2003. 238 s. ISBN 80-7302-049-1.
- TALANDOVÁ, P. *Přístupy ve zpracování tabulek v systémech DTP*. Diplomová práce. Mendelova zemědělská a lesnická univerzita v Brně, 2006.
- TANTAU, T. AJ. *The TikZ and PGF Packages. Manual for version 2.10*. Vytvořeno: 2010-10-25. Dostupné v instalaci distribuce TeX Live 2012. Z příkazového řádku příkazem `texdoc pgf` nebo `texdoc tikz`.

Přílohy

A Seznam dokumentů použitých v analýze

- ABB. *Bezpečnostní relé. Řada C57x. Technické údaje* [online]. [cit. 2013-05-03]. Dostupné z:
[http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/eee06eb77034a807c125742f0042b37e/\\$file/rozs1.pdf](http://www05.abb.com/global/scot/scot209.nsf/veritydisplay/eee06eb77034a807c125742f0042b37e/$file/rozs1.pdf).
- ANON. *Celkový přehled povolených výrobců léčivých přípravků a kontrolních laboratoří v ČR ke dni 31.01.2013* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.sukl.cz/file/73988_1_1.
- ANON. *Ceník služeb poskytovaných ve spolupráci s jinými operátory veřejných telekomunikačních sítí* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.viahome.cz/cz/download/viaPhone_cenik_operatori.pdf.
- ANON. *[Ceník zkoušek]* [online]. [cit. 2013-05-03]. Dostupné z:
<http://www.dekra-automobil.cz/doc/tuning/stltuning-cenik.pdf>.
- ANON. *LIST OF BENEFICIARIES. SEZNAM PŘÍJEMCŮ PODPORY Z FONDŮ EU* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.msmt.cz/file/15621_17_1/.
- ANON. *Nabídka kurzů CŽV na FLKŘ pro akademický rok 2011/2012* [online]. [cit. 2013-05-03]. Dostupné z: http://www.utb.cz/file/35021_1_1/download.
- ANON. *Seznam recenzovaných neimpaktovaných periodik vydávaných v České republice (po aktualizaci roku 2010)* [online]. [cit. 2013-05-03]. Dostupné z:
<http://www.vyzkum.cz/storage/att/56CD95E5453F0A261028BBC8262692AF/Seznam%202010%20aktualizace%20k%2019.8.2011.pdf>.
- ANON. *Tabulky velikostí obuvi Reebok* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.sportovniaerobik.cz/upload.cs/c/c0869bd4_0_obuvrbk.pdf.
- ANON. *[Telefonní seznam Městského úřadu Ivančice]* [online]. Poslední aktualizace: 2013-01-25. [cit. 2013-05-03]. Dostupné z:
<http://www.ivancice.cz/muivtel-seznam.pdf>.
- ANON. *[Výsledková tabulka]* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.celnik.eu/aktivity/201009_fotbal/vysledkova_tabulka.pdf.
- ANON. *VÝSLEDKOVÁ TABULKA DRUŽSTEV MUŽŮ KARLOVARSKÉ KRAJSKÉ LIGY DOSPĚLÝCH PRO ROK 2012. VÝSLEDKOVÁ TABULKA DRUŽSTEV ŽEN KARLOVARSKÉ KRAJSKÉ LIGY DOSPĚLÝCH PRO ROK 2012* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.krajskaliga.cz/docs/2012/tabulka_mnichov.pdf.
- ANON. *ZEMĚDĚLSKÝ PŮDNÍ FOND, ROSTLINNÁ VÝROBA* [online]. [cit. 2013-05-03]. Dostupné z:
[http://www.czso.cz/csu/2012edicniplan.nsf/t/62002E535C/\\$File/212712k08.pdf](http://www.czso.cz/csu/2012edicniplan.nsf/t/62002E535C/$File/212712k08.pdf).
- CITROËN ČESKÁ REPUBLIKA S.R.O. *TECHNICKÉ PARAMETRY. CITROËN C1* [online]. [cit. 2013-05-03]. Dostupné z:
http://www.citroen.cz/Resources/Content/CZ/07_PDF_gallery/

- 04_techparam_VP/Technicke_parametry_Citroen_C1.pdf.
ČESKÁ TISKOVÁ KANCELÁŘ. *proPŘEHLED. Ceník služby platný pro rok 2012* [online]. [cit. 2013-05-03]. Dostupné z:
http://i4.cn.cz/ctk/proprehled/Cenik_sluzby_PROprehled_2012.pdf.
ČESKOMORAVSKÝ ŠTĚRK, A.S., KAMENIVO SLOVAKIA, A.S. ÚSEK PRODEJE
ČMŠ. *Ceníky 2013. CENÍK KAMENIVA* [online]. [cit. 2013-05-03]. Dostupné
z: http://www.heidelbergcement.com/NR/rdonlyres/5677D40D-F277-42CD-9F82-B05DBAB38687/0/Cen%C3%ADk2013_0110.pdf.
DISTR. *CENÍK SEŠITŮ – OBJEDNÁVKA* [online]. [cit. 2013-05-03]. Dostupné
z: <http://www.distr.cz/soubory/sesity.pdf>.
FIO BANKA, A.S. *CENÍK FINANČNÍCH OPERACÍ A SLUŽEB* [online]. Praha,
2013-04-12. [cit. 2013-05-03]. Dostupné z:
https://www.fio.cz/docs/cz/cenik_bankovni_sluzby.pdf.
HYUNDAI. *HYUNDAI i40. Ceník osobních vozů (platnost od 1. května 2013).*
Typ karosérie: kombi [online]. [cit. 2013-05-03]. Dostupné z:
<http://www.hyundai.cz/data/files/pdf/cenik/cenik-i40-kombi.pdf>.
INSPORTLINE. FITNESS EQUIPMENT. *Uživatelský manuál – CZ. IN 4096*
Multifunkční posilovací lavice inSPORTline Bastet [online]. [cit. 2013-05-03].
Dostupné z:
http://obchod.ronnie.cz/_img/doc/2011-12-27-bastet-navod.pdf.
MŠ DA VINCI. *Přihláška do MŠ da Vinci (pobočka Zvole)* [online]. [cit.
2013-05-03]. Dostupné z:
http://www.msdavinci.cz/wp-content/uploads/2012/08/prihlaska_MSdaVinci_2012_2013_Zvole.doc.
PLASTIKA SV S.R.O. *Ceník ventilátorů NV a náhradních dílů. Platný od 1.1.*
2013 [online]. [cit. 2013-05-03]. Dostupné z:
<http://www.plastika-sv.cz/cenik-nv.pdf>.
STAŠA. [*Ceník. Plovoucí podlahy*] [online]. [cit. 2013-05-03]. Dostupné z:
http://www.stasa.cz/PDF/cenik_plovouci_podlahy_stasa.pdf.
ŠKODA AUTO. *Ceník. ŠKODA Fabia* [online]. [cit. 2013-05-03]. Dostupné z:
<http://www.skoda-auto.cz/sitecollectiondocuments/skoda-auto/ke-stazeni/fabia-cenik.pdf>.
TER HÜRNE. *Laminátové podlahy – formát, který je trendy. 6 atraktivních dekorů*
– lamely o délce 2 m [online]. [cit. 2013-05-03]. Dostupné z:
http://www.au-mex.cz/files/ceniky/laminat_terhurne_10_x_192_x_2000_mm_3.pdf.
TRNEX, SPOL. S.R.O. *Ceník přírodní dlažby a obkladu FOSIVAL pro rok 2013*
[online]. [cit. 2013-05-03]. Dostupné z:
http://www.trnex.cz/img/down_soubor1001.pdf.
TŘEŠTÍKOVÁ, J. *INFORMACE PRO PŘIJATÉ STUDENTY* [online]. [cit.
2013-05-03]. Dostupné z: <http://www.gekom.cz/files/1015MGV.doc>.
VELUX ČESKÁ REPUBLIKA, S.R.O. *Ceník*. [online]. [cit. 2013-05-03]. Dostupné z:
http://www.hermescv.cz/cenik_velux/cenik.pdf.

- VYKLIČKÝ, M. *Ceník služeb UPC Česká republika, s.r.o.* [online]. Praha, 2013-04-26. [cit. 2013-05-03]. Dostupné z:
http://www.upc.cz/pdf/Cenik_telefon.pdf.
- ZÁKLADNÍ UMĚLECKÁ ŠKOLA TRUTNOV. *ZÁPIS KE STUDIU PRO ŠKOLNÍ ROK 20__/20__ PRO STÁVAJÍCÍ ŽÁKY ŠKOLY* [online]. [cit. 2013-05-03]. Dostupné z: http://www.zustrutnov.cz/administrace_kestazeni/data/Z%C3%81PISOV%C3%9D%20LIST%20%28pro%20st%C3%A1vaj%C3%AD%C3%AD%20%C5%BE%C3%A1ky%20%C5%A1koly%29.pdf.
- ZEMĚMĚŘICKÝ ÚŘAD. *CENÍK PRODUKTŮ A VÝKONŮ ZEMĚMĚŘICKÉHO ÚŘADU* [online]. [cit. 2013-05-03]. Dostupné z:
<http://geoportal.cuzk.cz/Dokumenty/Cenik.pdf>.

B CD příloha

Na CD přiloženém k této diplomové práci jsou umístěny všechny tři soubory stylu:

- `tableenvdefaults.sty`
- `tableenvdefinitions.sty`
- `tableenvironment.sty`