

Katedra informatiky
Přírodovědecká fakulta
Univerzita Palackého v Olomouci

BAKALÁŘSKÁ PRÁCE

Editor Petriho Sítí



2019

Vedoucí práce: Mgr. Petr Osička,
Ph.D.

Roman Wehmhöner

Studijní obor: Aplikovaná informatika,
prezenční forma

Bibliografické údaje

Autor:	Roman Wehmhöner
Název práce:	Editor Petriho Sítí
Typ práce:	bakalářská práce
Pracoviště:	Katedra informatiky, Přírodovědecká fakulta, Univerzita Palackého v Olomouci
Rok obhajoby:	2019
Studijní obor:	Aplikovaná informatika, prezenční forma
Vedoucí práce:	Mgr. Petr Osička, Ph.D.
Počet stran:	30
Přílohy:	1 CD/DVD
Jazyk práce:	český

Bibliographic info

Author:	Roman Wehmhöner
Title:	Petri Nets Editor
Thesis type:	bachelor thesis
Department:	Department of Computer Science, Faculty of Science, Palacký University Olomouc
Year of defense:	2019
Study field:	Applied Computer Science, full-time form
Supervisor:	Mgr. Petr Osička, Ph.D.
Page count:	30
Supplements:	1 CD/DVD
Thesis language:	Czech

Anotace

Cílem bakalářské práce bylo vytvořit editor petriho sítí umožňující jednoduché a pohodlné ovládání. Editor také obsahuje základní nástroje pro analýzu petriho sítí.

Synopsis

Klíčová slova: styl textu; závěrečná práce; dokumentace; ukázkový text

Keywords: text style; thesis; documentation; sample text

Děkuji, děkuji, děkuji.

Místopřísežně prohlašuji, že jsem celou práci včetně příloh vypracoval/a samostatně a za použití pouze zdrojů citovaných v textu práce a uvedených v seznamu literatury.

datum odevzdání práce

podpis autora

Obsah

1	Petriho síť	8
1.1	Základní definice	8
1.2	Vizuální zobrazení sítě	9
1.3	Využití	9
1.4	Graf dosažitelnosti	10
1.4.1	Definice	10
1.4.2	Vlastnosti odvoditelné z Grafu dosažitelnosti	10
1.4.3	Příklady grafu dosažitelnosti	10
1.5	Graf pokrytí	10
1.5.1	Sestrojení grafu	11
1.5.2	Různé výsledky grafu pokrytí	11
1.5.3	Upravená verze vlastností	11
1.6	Příklady sítí	13
2	Editor	13
2.1	Systémové požadavky	13
2.2	Ovládání	13
2.2.1	Postranní panel	13
2.2.2	Hlavní plocha editoru	13
2.2.3	Panel nástrojů editoru	13
2.2.4	Tabulka ohodnocení	13
2.2.5	Výsledky analýzy	13
2.2.6	Tisk sítě	13
2.2.7	Klávesové zkratky	13
3	Použité technologie	13
3.1	nodejs	13
3.2	Typescript	13
3.3	Electron	14
3.4	Javascriptová Knihovna Data driven documents (D3)	14
3.5	Scalable Vector Graphics (SVG)	14
4	Stavba programu	14
4.1	Třída 1	14
4.2	Třída 2	14
4.3	Třída 3	14
4.4	Třída 4	14
5	Styly pro psaní bakalářských a diplomových prací	14
5.1	Požadavky a podprovaná prostředí	15
5.2	Přepínače	15
5.3	Geometrie stránky	15

6	Sazba částí dokumentu	17
6.1	Sazba úvodní strany či obsahu	17
6.2	Závěry	17
6.3	Matematika	17
6.4	Sazba literatury	18
6.4.1	Sazba bibliografie přes BIB _Λ T _E X	18
6.4.2	Manuální sazba bibliografie	18
6.5	Drobná makra	18
6.6	Sazba rejstříku	20
6.7	Sazba zdrojových kódů	20
	Závěr	23
	Conclusions	24
	A První příloha	25
	B Druhá příloha	25
	C Obsah přiloženého CD/DVD	25
	Seznam zkratk	27
	Literatura	28
	Rejstřík	30

Seznam obrázků

Seznam tabulek

1	Seznam přepínačů	16
2	Seznam přepínačů	19

Seznam vět

1	Definice (Název definice)	20
	Důkaz (Název důkazu)	20
2	Poznámka (Pumpovací věta)	20
3	Příklad (Pumpovací věta)	21
4	Lemma (Název definice)	21
5	Důsledek (Název důkazu)	21
6	Věta (Pumpovací věta)	21

Seznam zdrojových kódů

1	Volání třídy kidiplom	15
2	Sazba závěrů	17
3	C++	21
4	JS	21
5	C#	21
6	SQL	22
7	TutorialD	22

TODO: Smazat todo: command

1 Petriho síť

Táto kapitola byla inspirovaná a čerpala informace z Understanding petri nets[1]

1.1 Základní definice

Petriho síť jsou matematickým nástrojem pro modelování a simulaci paralelních procesů a jejich synchronizaci. Jsou tvořené místy, přechody a hranami které jsou vždy propojením jednoho místa s jedním přechodem.

$$N = \langle P, T, A, M_0 \rangle$$

- N je Petriho síť
- P je konečná množina míst
- T je konečná množina přechodů
- A je konečná množina hran $A \subseteq ((P \times T) \cup (T \times P)) \times \mathbb{N}_0$
kde číslo symbolizuje násobek kolik značek hrana 'přesune'
- $M_0 : P \rightarrow \mathbb{N}_0$ je počáteční ohodnocení sítě (zkráceně ohodnocení) míst kde pro každé místo $p \in P$ existuje počet jeho značek $m \in M_0$

Pro odkazování na jednotlivé členy prvků z množiny hran budeme používat notaci $P(a)$ pro odkázání na místo v hraně $a \in A$, $T(a)$ pro odkázání na přechod a $AM(a)$ pro odkaz na násobek.

Každý přechod t může mít 'přiřazený' libovolný počet hran, kde každá hrana a je propojením s některým z míst $p \in P$.

Hrany přechodu t můžeme rozlišit na hrany směřující do přechodu

$$\rightarrow t = \{a \in A \mid a \in (P \times T \times \mathbb{N}_0) \wedge t = T(a)\}$$

a hrany směřující z přechodu (do místa)

$$t \rightarrow = \{a \in A \mid a \in (T \times P \times \mathbb{N}_0) \wedge t = T(a)\}$$

dohromady pak všechny hrany přechodu t jsou spojením těchto dvou množin

$$ArcesOfTransition(t, A) = (\rightarrow t \cup t \rightarrow) \subset A$$

Aktuální stav petriho sítě neboli ohodnocení M je funkce přiřazující každému místu $p \in P$ petriho sítě počet značek

$$M(p) \in \mathbb{N}_0$$

počáteční ohodnocení petriho sítě se značí M_0 .

Pro dané ohodnocení M je přechod $t \in T$ označený jako povolený pokud všechny hrany směřující do přechodu $\rightarrow t$ splňují svou podmínku tzn. hrana splňuje svoji podmínku pokud místo ze kterého vychází má vyšší nebo stejné ohodnocení (v daném M) než je násobek hrany AM

$$IsEnabled(P, t, A, M) = (\forall a \in \rightarrow t) M(P(a)) \geq AM(a)$$

Pak si můžeme ještě definovat množinu všech povolených přechodů pro zadané ohodnocení

$$EnabledTransitions(P, T, A, M) = \{t \in T | IsEnabled(P, t, A, M)\}$$

Pokud je přechod t v ohodnocení M Petriho sítě **povolen**, znamená to že může dojít k aktivování tohoto přechodu čímž dojde ke změně aktuálního ohodnocení z M do ohodnocení M' tak, že pro každé místo $p \in P$ a každou hranu $A_{pt} \subset ArcsOfTransition(t, A)$ spojující p s t že nové ohodnocení v místě $M'(p)$ je sumou násobků hran $\sum_{a \in A_{pt}} AM(a)$ a původního hodnocení $M(p)$

$$FireTransition(P, t, A, M) = function M'$$

Výsledné ohodnocení M' je pak definováno

$$M'(p) = M(p) + \sum_{a \in \{a_{tp} \in ArcsOfTransition(t, A) \mid P(a_{tp})=p\}} AM(a)$$

Tuto změnu můžeme značit $M \rightarrow^t M'$

Ohodnocení M' je označené jako **dosazitelné**^{1} z ohodnocení M pokud existuje sekvence přechodů taková, že jejich postupným aktivováním z ohodnocení M vznikne ohodnocení M' . Ohodnocení M' je dostupné z ohodnocení M pak značíme $M \rightarrow^* M'$.

TODO: testovací hrany

1.2 Vizuální zobrazení sítě

TODO: obrázky

1.3 Využití

Petriho sítě se používají k analýze a modelování paralelních a distribuovaných systému, databázových systémů atd. a to až už pro analýzu při vývoji softwaru a nebo pro popis vnitřní struktury již hotového proprietárního softwaru umožňující lepší porozumění uživateli.

1.4 Graf dosažitelnosti

Graf dosažitelnosti je jeden z nejzákladnějších nástrojů pro analýzu Petriho sítí. Obsahuje vždy počáteční ohodnocení a všechny ohodnocení které jsou dostupné z počátečního ohodnocení, takovéto ohodnocení budeme nazývat **dosažitelné ohodnocení**^{2}. Vrcholy grafu jsou jednotlivá ohodnocení a hrany grafu jsou značené přechody které jsou aktivované aby z počátečního ohodnocení vzniklo cílové.

1.4.1 Definice

$$RG = \{M, \langle M, t, M' \rangle\}$$

- RG je Graf dosažitelnosti
- M je Vrchol grafu který je zároveň konkrétní ohodnocení petriho sítě
- $\langle M, t, M' \rangle$ je Hrana grafu která je změnou z hodnocení M přechodem t ze kterého vzniká M'

1.4.2 Vlastnosti odvoditelné z Grafu dosažitelnosti

Z grafu dosažitelnosti Petriho sítě jsou odvoditelné tyto vlastnosti:

Petriho síť **skončí**^{3} za předpokladu že graf je konečný a zároveň neobsahuje žádné cykly. Neboli Petriho síť vždy po nějakém počtu kroků dojde do stavu kdy žádný přechod není povolený.

Petriho síť je **bez mrtvého bodu**^{4} pokud z každého vrcholu grafu vede alespoň jedna hrana. Petriho síť má v každém ohodnocení povolený alespoň jeden přechod.

Petriho síť je **slabě živá**^{5} pokud pro každý přechod existuje v grafu hrana označená tímto přechodem. Pro každý přechod Petriho sítě můžeme najít sekvenci přechodů (i prázdnou) takovou že z počátečního ohodnocení dojdeme do ohodnocení které povoluje daný přechod.

Petriho síť je **živá**^{6} pokud je její graf silně souvislý. Stejně jako **slabě živá** s tím rozdílem že musí splňovat vlastnost ze všech dosažitelných ohodnocení a ne jen z počátečního ohodnocení ale ze všech dosažitelných ohodnocení Petriho sítě.

1.4.3 Příklady grafu dosažitelnosti

1.5 Graf pokrytí

Hlavní nevýhodou grafu dosažitelnosti je že může být nekonečný a tudíž je nemožné ho zkonstruovat celý. Můžeme velice jednoduše navrhnout a sestavit triviální Petriho síť u které by konstrukce jejího grafu dosažitelnosti nikdy neskončila.

TODO: obrázek petriho sítě kde je jedné place a jedna transition a hrana z transition do place

Proto existuje rozšířená verze grafu dosažitelnosti která může obsahovat tzv. ω ohodnocení které mimo celých čísel přiřadí alespoň jednomu místu i hodnotu ω symbolizující že místo může nabívat nekonečně vysokého počtu značek. Petriho síť se nemůže nacházet v ω ohodnocení, toto ohodnocení je pouze pro vytvoření abstrakce v grafu pokrytí.

Protože hodnotu ω bereme jako nekonečno pak od ní můžeme odečíst nebo přičíst libovolně velké číslo a hodnota se nezmění.

$$\dots = \omega - 2 = \omega - 1 = \omega = \omega + 1 = \omega + 2 = \dots$$

Ohodnocení M značíme jako že je ostře menší $<$ než ohodnocení M' pokud pro každé místo p platí $M(p) \leq M'(p)$ a alespoň pro jedno místo p platí $M(p) < M'(p)$.

$$M < M' = ((\forall p \in P) M(p) \leq M'(p)) \wedge (\exists p \in P) M(p) < M'(p)$$

1.5.1 Sestrojení grafu

Sestrojování grafu probíhá postupně přidáváním hran. Nejdříve se přidá počáteční ohodnocení jako kořen grafu. Následně se z grafu vybírají náhodně nevypočítané povolené přechody a pokud vedou do místa které ještě v grafu není tak se přidá a pokud je ostře menší než ohodnocení ze kterého je dosažitelné tak se přidají ω hodnoty na místa ve kterých má více značek. Algoritmus končí výpočet až jsou všechny povolené přechody pro všechny vrcholy v grafu vypočítané.

Sestrojení grafu pseudokód [MakeCoverabilityGraph1](#)

Pokud sestrojený graf pokrytí neobsahuje žádné ω ohodnocení, pak je graf totožný s grafem dosažitelnosti. Pokud graf pokrytí obsahuje ω ohodnocení, znamená to že graf dosažitelnosti by byl nekonečný a tudíž by nebylo možné ho zkonstruovat celý a nešli by na něm zjišťovat některé nebo všechny vlastnosti. Proto si vystačíme s algoritmem na vytváření grafu pokrytí.

1.5.2 Různé výsledky grafu pokrytí

Při konstrukci grafu pokrytí záleží v jakém pořadí se hrany přidávají a výsledný graf může mít různý počet vrcholů a hran v závislosti na pořadí přidávání hran. V našem algoritmu využíváme funkce *RandomElement* která vybere náhodný prvek množiny. Pokud bychom chtěli sestrojit minimální graf pokrytí, museli bychom nahradit funkci *RandomElement* nějakou nedeterministickou funkcí která by vždy vybrala přechody právě v takovém pořadí aby došlo k sestrojení minimálního grafu.

TODO: Příklad obrázek

1.5.3 Upravená verze vlastností

Oproti grafu dosažitelnosti náš graf pokrytí tak jak jsme ho sestrojili pomocí algoritmu *MakeCoverabilityGraph* **TODO: odkaz**, protože mohou chybět

Algorithm 1 MakeCoverabilityGraph

```
1: function MAKECOVERABILITYGRAPH( $\langle P, T, A, M_0 \rangle$ )
2:    $\langle V, E, v_0 \rangle := \langle \{M_0\}, \emptyset, M_0 \rangle$ 
3:    $WorkSet := \emptyset$ 
4:   for all  $t \in EnabledTransitions(P, T, A, M_0)$  do
5:      $WorkSet := WorkSet \cup \{\langle M_0, t \rangle\}$ 
6:   end for
7:   while  $WorkSet \neq \emptyset$  do
8:      $\langle M, t \rangle := RandomElement(WorkSet)$ 
9:      $WorkSet := WorkSet \setminus \{\langle M, t \rangle\}$ 
10:     $M' := FireTransition(P, t, A, M)$ 
11:    for all  $\{M'' \mid M'' \in V \wedge (M'' \rightarrow^* M \vee M'' = M) \wedge M'' < M'\}$  do
12:      for all  $p \in P$  do
13:         $mp := M(p)$ 
14:        if  $M''(p) < M'(p)$  then
15:           $M'(p) := \omega$ 
16:        end if
17:      end for
18:    end for
19:    if  $M' \notin V$  then
20:       $V := V \cup \{M'\}$ 
21:      for all  $t \in EnabledTransitions(P, T, A, M')$  do
22:         $WorkSet := WorkSet \cup \{\langle M', t \rangle\}$ 
23:      end for
24:    end if
25:     $E := E \cup \{\langle M, t, M' \rangle\}$ 
26:  end while
27:  return  $\langle V, E, v_0 \rangle$ 
28: end function
```

1.6 Příklady sítí

TODO: síť + analýza

2 Editor

2.1 Systémové požadavky

TODO: vyžaduje myš

2.2 Ovládání

TODO: obrázek rozložení editoru

2.2.1 Postranní panel

TODO: jiné pojmenování ?

2.2.2 Hlavní plocha editoru

2.2.3 Panel nástrojů editoru

2.2.4 Tabulka ohodnocení

2.2.5 Výsledky analýzy

2.2.6 Tisk sítě

TODO: obrázek jak vytisknotu do PDF

2.2.7 Klávesové zkratky

3 Použité technologie

3.1 nodejs

Aplikace je psaná za pomoci opensourcové technologie nodeJS, která umožňuje využívat jazyk JavaScript pro psaní serverových aplikací. Cílem platformy nodeJS je vytvořit ekosystém pro jednodušší vývoj webových stránek a aplikací kde stačí pro vyrvaření funkcionality pouze jeden programovací jazyk.

3.2 Typescript

Typescript je opensource programovací jazyk od společnosti Microsoft který je nadstavbou nad jazykem JavaScript. Jelikož je Typescript nadstavbou nad programovacím jazykem JavaScript tak je jakýkoliv validní kód v JavaScriptu automaticky validním kódem v Typescriptu. Typescript se kompiluje do Javascriptu a

proto po stránce funkcionality nenabízí nic navíc avšak po stránce vývoje nabízí možnost statické typové kontroly se kterou je spjaté fungování našeptávačů v dnešních textových editorech a také nabízí možnost kompilace do starších verzí JavaScript se simulací funkcionality novějších verzí JavaScriptu. **TODO: příklady kódu ?**

3.3 Electron

Electron je opensource framework pro vytváření desktopových aplikací pomocí webových technologií JavaScript, HTML a CSS. Využívá NodeJS

3.4 Javascriptová Knihovna Data driven documents (D3)

TODO: příklady kódu ?

3.5 Scalable Vector Graphics (SVG)

TODO: příklady kódu ?

4 Stavba programu

4.1 Třída 1

4.2 Třída 2

4.3 Třída 3

4.4 Třída 4

TODO: pokračování ukázkového textu

Upozornění: Následující text dokumentace stylu, vyjma přílohy C, je rozpracovaná a (značně) neúplná verze!!!

5 Styly pro psaní bakalářských a diplomových prací

Toto jsou styly pro psaní bakalářských a diplomových prací přes typografický systém L^AT_EX, tedy **kistyles**.

5.1 Požadavky a podprovaná prostředí

Sada balíku **kistyles** podporuje následující distribuce systému \LaTeX :

- \TeX Live.

Jsou podporovány všechny výstupní ovladače, tedy jak **dvi**, tak **pdf** i **ps**. Funkčnost zmiňovaných distribucí byla ověřena na několika operačních systémech, mezi které patří:

1. Windows 8.1,
2. Archlinux,
3. Debian.

Důrazně se doporučuje používat aktuální verzi dané distribuce systému \LaTeX .

5.2 Přepínače

Styl kidiplom je z hlediska uživatele zastoupen ekvivalentně nazvanou třídou, kterou je třeba volat na začátku dokumentu:

```
1 \documentclass[
2   master=true,
3   font=sans,
4   printversion=false,
5   joinlists=true,
6   glossaries=true,
7   figures=true,
8   tables=true,
9   sourcecodes=true,
10  theorems=true,
11  bibencoding=utf8,
12  language=czech,
13  encoding=utf8,
14  field=inf,
15  index=true,
16  biblatex=true
17 ]{kidiplom}
```

Zdrojový kód 1: Volání třídy **kidiplom**

Následuje přehled přepínačů, je vždy uvedeno jméno přepínač, včetně výchozí hodnoty. Přepínače uvádí tabulka [1](#).

5.3 Geometrie stránky

Tento styl používá list velikosti A4. Pro sazbu prací je třeba použít jednostrannou sazbu. Levý okraj je rozšířen s ohledem na vazbu výsledné knižní podoby práce.

Tabulka 1: Seznam přepínačů

Přepínač	Výchozí hodnota	Popis
master	<code>false</code>	Povolí nebo zakáže režim diplomové práce. Výchozí režim je tedy bakalářská práce.
field	<code>ainfp</code>	Specifikuje studijní obor: ainf Aplikovaná informatika – prezenční, ainfk Aplikovaná informatika – kombinovaná, inf Informatika – prezenční, infv Informatika ve vzdělávání – kombinovaná, binf Bioinformatika – prezenční.
font	<code>serif</code>	Zapne či vypne podporu pěkného bezpatkového fontu. Možné hodnoty jsou: sans Bezpatkové písmo (písmo Iwona). serif Patkové písmo (písmo Computer Modern).
encoding	<code>utf8</code>	Kódování souboru dokumentu, doporučuje se ponechat výchozí hodnotu.
bibencoding	<code>utf8</code>	Kódování souboru bibliografie. Tato volba má smysl pouze, pokud je použita bibliografie skrze balíček BIB _Λ T _E X.
language	<code>czech</code>	Jazyk práce.
printversion	<code>false</code>	Je-li zapnuto, pak budou odkazy vysázeny optimalizovaně pro knižní sazbu. Tuto volbu je nutno použít pro tisk práce.
joinlists	<code>true</code>	Je-li zapnuto, pak seznamy obrázků, tabulek, vět a zdrojových kódů sázené za obsahem nebudou rozděleny na samostatné stránky.
figures	<code>true</code>	Je-li zapnuto, pak v seznamech položek bude zahrnut seznam obrázků.
tables	<code>true</code>	Je-li zapnuto, pak v seznamech položek bude zahrnut seznam tabulek.
theorems	<code>false</code>	Je-li zapnuto, pak v seznamech bude zahrnut seznam teorémů.
sourcecodes	<code>false</code>	Je-li zapnuto, pak v seznamech bude zahrnut seznam zdrojových kódů.
glossaries	<code>false</code>	Je-li zapnuto, pak na konci dokumentu bude vysázen seznam zkratk.
index	<code>false</code>	Zapíná podporu sazby rejstříku.
biblatex	<code>true</code>	Zapne sazbu bibliografie přes balík BIB _Λ T _E X.

6 Sazba částí dokumentu

6.1 Sazba úvodní strany či obsahu

Vysázení všech podstatných částí úvodu práce obstará makro `\maketitle`. Pro správné vysázení všech částí a meta-informací je potřeba použít makra `\title`, `\author` a další. Jejich přehled lze najít ve zdrojovém souboru tohoto dokumentu. V případě použití **pdf** výstupu se generuje i dodatečná hlavička souboru s meta-informacemi jako je autor dokumentu, název práce či dalšími.

6.2 Závěry

Závěr práce by se měl poskytnout jak v původním jazyce práce, tak v jazyce anglickém. Pro sazbu závěru jsou k dispozici příslušná makra. Berte na vědomí, že v anglickém závěru se aktivuje plně anglická sazba se všemi konvencemi. Tedy je třeba používat anglické uvozovky a další správné typografické prvky.

```
1 % Tiskne český závěr práce.
2 \begin{kiconclusions}
3   Závěr práce v \uv{českém} jazyce.
4 \end{kiconclusions}
5
6 % Tiskne anglický závěr práce.
7 \begin{kiconclusions}[english]
8   Thesis conclusions written in \uv{English}.
9 \end{kiconclusions}
```

Zdrojový kód 2: Sazba závěrů

6.3 Matematika

Pro sazbu matematiky je k dispozici sada standardních maker.

$$\langle f \rangle, [g], [h], \lceil i \rceil$$

$$A_{m,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{pmatrix}$$

$$M = \begin{bmatrix} \frac{5}{6} & \frac{1}{6} & 0 \\ \frac{5}{6} & 0 & \frac{1}{6} \\ 0 & \frac{5}{6} & \frac{1}{6} \end{bmatrix}$$

6.4 Sazba literatury

Pro sazbu literatury má uživatel dvě možnosti. Může použít služeb balíků `BIBLATEX`, který je pro `kistyles` zapnutý, či lze použít manuální sazbu bibliografie.

6.4.1 Sazba bibliografie přes BibL^AT_EX

Při použití tohoto balíku se data o použité literatuře ukládají do dedikovaného textového souboru, ukázkou najdete i v tomto stylu pod jménem `bibliografie.bib`.

Formát daného souboru je nad rámec této dokumentace a je na každém uživateli, aby si jej nastudoval. Bibliografie se tiskne makrem `\printbibliography`. Taktéž v preambuli dokumentu je třeba definovat, který soubor data bibliografie obsahuje, tedy například `\bibliography{bibliografie.bib}`.

Dokument, který využívá `BIBLATEX` je následně nutné přeložit jak pomocí překladače zvoleného ovladače, tak pomocí aplikace `biber`. Více informací poskytnou soubor `Makefile` z distribuce tohoto stylu.

Výhodou tohoto přístupu je, že bibliografie se vysází automaticky a (obvykle) není třeba manuální úprava formátování.

6.4.2 Manuální sazba bibliografie

Manuální sazba obnáší vysazení prostředí `thebibliography` ručně. To je nad rámec tohoto dokumentu. Ukázkou tohoto přístupu lze samozřejmě nalézt ve zdrojovém souboru tohoto dokumentu nebo také [zde](#).

Pro aktivaci manuální sazby bibliografie je třeba volat třídu `kidiplom` s parametrem `biblatex=false`. Mějte, prosím, na paměti, že v tomto módu jsou makra `\bibliography` a `\printbibliography` nedostupná.

6.5 Drobná makra

Základní styl definuje hned několik maker pro usnadnění práce. Například makro `\buno` vysází řetězec „bez újmy na obecnosti“. Je k dispozici i verze s prvním velkým písmenem, `\Buno`.

Je rovněž možno přidávat položky do seznamu zkratk. K tomu slouží makro `\newacronym`, které lze použít například jednoduše jako `\newacronym{UPOL}{UPOL}{\kitextunivcz}`. Na danou zkratku se pak lze odkazovat jednoduše, `\gls{UPOL}`.

Sazba uvozovek respektuje nastavení částí dokumentu, a proto se doporučuje používat makro `\uv`. V anglické závěru práce toto platí taky, viz tato PDF ukáзка.

Styl podporuje sazbu odstavců v tabulkách, více obsahuje tabulka [2](#).

K dispozici jsou také makra pro sazbu C# (`\csharp`) či C++ (`\cpp`).

Tabulka 2: Seznam přepínačů

Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.	Etiam suscipit aliquam arcu. Aliquam sit amet est ac purus bibendum congue. Sed in eros. Morbi non orci. Pellentesque mattis lacinia elit. Fusce molestie velit in ligula. Nullam et orci vitae nibh vulputate auctor. Aliquam eget purus. Nulla auctor wisi sed ipsum. Morbi porttitor tellus ac enim. Fusce ornare. Proin ipsum enim, tincidunt in, ornare venenatis, molestie a, augue. Donec vel pede in lacus sagittis porta. Sed hendrerit ipsum quis nisl. Suspendisse quis massa ac nibh pretium cursus. Sed sodales. Nam eu neque quis pede dignissim ornare. Maecenas eu purus ac urna tincidunt congue.	Etiam pede massa, dapibus vitae, rhoncus in, placerat posuere, odio. Vestibulum luctus commodo lacus. Morbi lacus dui, tempor sed, euismod eget, condimentum at, tortor. Phasellus aliquet odio ac lacus tempor faucibus. Praesent sed sem. Praesent iaculis. Cras rhoncus tellus sed justo ullamcorper sagittis. Donec quis orci. Sed ut tortor quis tellus euismod tincidunt. Suspendisse congue nisl eu elit. Aliquam tortor diam, tempus id, tristique eget, sodales vel, nulla. Praesent tellus mi, condimentum sed, viverra at, consectetur quis, lectus. In auctor vehicula orci. Sed pede sapien, euismod in, suscipit in, pharetra placerat, metus. Vivamus commodo dui non odio. Donec et felis.
---	--	--

6.6 Sazba rejstříku

Sazba rejstříku sestává z několika kroků:

1. Je třeba přes volbu `index=true` rejstříkování povolit.
2. Použitím makra `\index` rejstříkovat vybrané pojmy.
3. Kompilovat s použitím utility `makeindex`. Pro specifika tohoto kroku si stačí prohlédnout soubor `Makefile`.

Makro `\index` je redefinováno tak, že sází klikací odkaz na výraz v rejstříku. Je doporučeno jej použít ihned za výrazem^{[{7}](#)}.

Omezení redefinovaného makra `\index`: klikací odkaz nefunguje, pokud použijete konstrukci `\index{výraz|makro}` (resp. `\index{výraz|(makro)}`), např. `\index{výraz|textit}`.

Rejstřík lze vysázet pomocí makra `\printindex`.

6.7 Sazba zdrojových kódů

Styl nabízí dva způsoby sazby zdrojových kódů:

1. Sazbu řádkových kódů, například `background-color: white;`. K tomu slouží makro formátu `\kiinlinecode{jazyk}{separátor}{kód}`. Za separátor je vhodné volit jakýkoliv znak, který se nevyskytuje v samotném sazeném zdrojovém kódu. Za jazyk je nutno dosadit jeden z těchto: C, TeX, PHP, HTML, Lisp, SQL, TeX, Python, Java, TutorialD, text, csharp, cpp, JavaScript, CSS.
2. Sazbu zdrojových kódů do separátních prostředí. Takto vytištěný kód se objeví v seznamu zdrojových kódů. Ukázka například zdrojový kód [3](#). Ukázku sazby naleznete ve zdrojovém kódu tohoto dokumentu.

Definice 1 (Název definice)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. **Univerzita Palackého v Olomouci**
(UPOL)

Důkaz (Název důkazu)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. □

POZNÁMKA 2 (PUMPOVACÍ VĚTA)

[illegible]

PŘÍKLAD 3 (PUMPOVACÍ VĚTA)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.

Lemma 4 (Název definice)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd.

Důsledek 5 (Název důkazu)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd.

Věta 6 (Pumpovací věta)

Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd. Abcd.
Abcd. Abcd. Abcd. Abcd. Abcd.

```
1  int main("cs acsa") // komentar
2  int main("cs acsa") // komentar
3  int main("cs acsa") // komentar
4  int main("cs acsa") // komentar
5  int main("cs acsa") // komentar
```

Zdrojový kód 3: C++

```
1  new object() // komentar
```

Zdrojový kód 4: JS

```
1  public static int main("cs acsa") // komentar
```

Zdrojový kód 5: C#

```
1  SELECT * FROM table_1; /* komentar */
```

Zdrojový kód 6: SQL

```
1  table_1 AND table_2;
```

Zdrojový kód 7: TutorialD

Závěr

Závěr práce v „českém“ jazyce.

Conclusions

Thesis conclusions in “English”.

A První příloha

Text první přílohy

B Druhá příloha

Text druhé přílohy

C Obsah přiloženého CD/DVD

Na samotném konci textu práce je uveden stručný popis obsahu přiloženého CD/DVD, tj. jeho závazné adresářové struktury, důležitých souborů apod.

bin/

Instalátor `INSTALATOR` programu, popř. program `PROGRAM`, spustitelné přímo z CD/DVD. / Kompletní adresářová struktura webové aplikace `WEBOVKA` (v ZIP archivu) pro zkopírování na webový server. Adresář obsahuje i všechny runtime knihovny a další soubory potřebné pro bezproblémový běh instalátoru a programu z CD/DVD / pro bezproblémový provoz webové aplikace na webovém serveru.

doc/

Text práce ve formátu PDF, vytvořený s použitím závazného stylu KI PřF UP v Olomouci pro závěrečné práce, včetně všech příloh, a všechny soubory potřebné pro bezproblémové vygenerování PDF dokumentu textu (v ZIP archivu), tj. zdrojový text textu, vložené obrázky, apod.

src/

Kompletní zdrojové texty programu `PROGRAM` / webové aplikace `WEBOVKA` se všemi potřebnými (příp. převzatými) zdrojovými texty, knihovnami a dalšími soubory potřebnými pro bezproblémové vytvoření spustitelných verzí programu / adresářové struktury pro zkopírování na webový server.

readme.txt

Instrukce pro instalaci a spuštění programu `PROGRAM`, včetně všech požadavků pro jeho bezproblémový provoz. / Instrukce pro nasazení webové aplikace `WEBOVKA` na webový server, včetně všech požadavků pro její bezproblémový provoz, a webová adresa, na které je aplikace nasazena pro účel testování při tvorbě posudků práce a pro účel obhajoby práce.

Navíc CD/DVD obsahuje:

data/

Ukázková a testovací data použitá v práci a pro potřeby testování práce při tvorbě posudků a obhajoby práce.

install/

Instalátory aplikací, runtime knihoven a jiných souborů potřebných pro provoz programu PROGRAM / webové aplikace WEBOVKA, které nejsou standardní součástí operačního systému určeného pro běh programu / provoz webové aplikace.

literature/

Vybrané položky bibliografie, příp. jiná užitečná literatura vztahující se k práci.

U veškerých cizích převzatých materiálů obsažených na CD/DVD jejich zahrnutí dovolují podmínky pro jejich šíření nebo přiložený souhlas držitele copyrightu. Pro všechny použité (a citované) materiály, u kterých toto není splněno a nejsou tak obsaženy na CD/DVD, je uveden jejich zdroj (např. webová adresa) v bibliografii nebo textu práce nebo v souboru `readme.txt`.

Literatura

- [1] REISIG, Wolfgang. *Understanding petri nets : modeling techniques, analysis methods, case studies*. Berlin: Springer, 2013. ISBN 978-3-642-33277-7.
- [2] BRATKOVÁ, Eva (sest.). *Metody citování literatury a strukturování bibliografických záznamů podle mezinárodních norem ISO 690 a ISO 690-2: metodický materiál pro autory vysokoškolských kvalifikačních prací* [online]. Verze 2.0, aktualiz. a rozšíř. Praha: Odborná komise pro otázky elektronického zpřístupňování vysokoškolských kvalifikačních prací, Asociace knihoven vysokých škol ČR, 2008 [cit. 2011-2-2]. 60 s. Dostupný z: <http://www.evskp.cz/SD/4c.pdf>.
- [3] BORGMAN, Christine L. *From Gutenberg to the global information infrastructure: access to information in the networked world*. First. Cambridge (Mass): The MIT Press, 2003. xviii, 324 s. ISBN 0-262-52345-0.
- [4] GREENBERG, David. Camel drivers and gatecrashers: quality control in the digital research library. In HAWKINS, B.L; BATTIN, P (ed.). *The mirage of continuity: reconfiguring academic information resources for the 21st century*. Washington (D.C.): Council on Library and Information Resources; Association of American Universities, 1998, s. 105–116.
- [5] LYNCH, C. Where do we go from here?: the next decade for digital libraries. *DLib Magazine* [online]. 2005, vol. 11, no. 7/8 [cit. 2005-8-15]. Dostupný z: <http://www.dlib.org/dlib/july05/lynch/07lynch.html>. ISSN 1082-9873.
- [6] NÁRODNÍ KNIHOVNA. A big paper. *The journal of big papers*. 1991, vol. 12, no. 3. ISSN 2232-332X.
- [7] DĚŤA, Hugh; RYCHLÍK, Tomáš. *A big paper: Podtitul* [online]. Druhé vyd. Praha: Academia, 1991 [cit. 2011-1-12]. 550 s. Pokusná edice. Dostupný z: <http://pokus.cz>. ISBN 978-44-55-X.
- [8] DĚŤA, Hugh; RYCHLÍK, Tomáš; DALŠÍ, Pepa aj. *Úplně úžasná knížka*. Třetí vyd., 1991.
- [9] DĚŤA, Hugh; RYCHLÍK, Tomáš; DALŠÍ, Pepa, et al. *Úplně úžasná knížka*. 3rd ed. Praha: MIT Press, 1991. 332 s.
- [10] FREELY, I.P. A small paper: Podtitulek. *The journal of small papers*. 1997, roč. 1, č. 3, s. 2–5. to appear.
- [11] JASS, Hugh. A big paper. *The journal of big papers*. 1991, roč. 23.
- [12] ČERNÝ, Hugh. Titulek. *The journal of big papers*. 1991, roč. 12, č. 2, s. 22–44. Dostupný také z: <http://dx.doi.org/10.112.22/jkn>.
- [13] KOLLMANNOVÁ, Ludmila; BUBENIKOVÁ, Libuše; KOPECKÁ, Alena. *Angličtina pro samouky*. 5. vyd. Praha: Státní pedagogické nakladatelství, 1977. 525 s. Učebnice pro samouky. ISBN 80-04-25663-5.
- [14] KOLLMANNOVÁ, Ludmila; BUBENIKOVÁ, Libuše; KOPECKÁ, Alena. *Angličtina pro samouky*. 5. vyd. NOVOTNÁ, Pepina. Praha: Státní pedagogické nakladatelství, 1977. 2. Podkapitola, s. 22–29. ISBN 80-04-25663-5.

- [15] *TUGBoat*. 1980-. Dostupný také z: <http://tugboat.tug.org/>. ISSN 1222-3333.
- [16] KNUTH, Donald. Journeys of T_EX. *TUGBoat*. 2003, vol. 17, no. 3, s. 12–22. Dostupný také z: <http://tugboat.tug.org/kkk.pdf>. ISSN 1222-3333.
- [17] GENIÁLŇÍ, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007. ISBN 978-222-626-222-2.
- [18] VLAŠTOVKA, Josef. Velmi zajímavý článek. In GENIÁLŇÍ, Jiří (ed.). *Mimořádně užitečný sborník*. Praha: Academia, 2007, s. 22–45. ISBN 978-222-626-222-2.

Rejstřík

dosažitelné ohodnocení, [9](#), [10](#)

sít živá, [10](#)

sít bez mrtvého bodu, [10](#)

sít skončí, [10](#)

sít slabě živá, [10](#)

výraz, [20](#)