



## ZADÁNÍ BAKALÁŘSKÉ PRÁCE

<b>Název:</b>	Sémantické sítě matematických znalostí
<b>Student:</b>	Ondřej Šodek
<b>Vedoucí:</b>	Ing. Tomáš Kalvoda, Ph.D.
<b>Studijní program:</b>	Informatika
<b>Studijní obor:</b>	Znalostní inženýrství
<b>Katedra:</b>	Katedra aplikované matematiky
<b>Platnost zadání:</b>	Do konce letního semestru 2019/20

### Pokyny pro vypracování

1. Prostudujte teorii sémantických sítí a případně další způsoby popisu vztahů mezi různými objekty a koncepty, zejména se soustředte na matematické domény, speciálně oblast matematických předmětů na FIT.
2. Proveďte rešerši různých metodik pro vytváření sémantických sítí. Prozkoumejte možné přístupy k odhalení a extrakci vztahů využívající strojovou analýzu textu.
3. Aplikujte znalosti z předchozích bodů na studijní text předmětu BI-ZMA, jeho zdrojový kód ve zpracovatelné podobě dodá vedoucí práce. Vytvořte sémantickou síť důležitých pojmů, vlastností, konceptů a vztahů mezi nimi.
4. Vytvořte sémantickou síť pro předmět BI-ZMA ve formě přehledné a snadno použitelné pro studenty. Zajistěte propojení se studijním textem BI-ZMA (jeho webovou verzí).

### Seznam odborné literatury

Dodá vedoucí práce.

Ing. Karel Klouda, Ph.D.  
vedoucí katedry

doc. RNDr. Ing. Marcel Jiřina, Ph.D.  
děkan

V Praze dne 6. února 2019





**FAKULTA  
INFORMAČNÍCH  
TECHNOLÓGIÍ  
ČVUT V PRAZE**

Bakalářská práce

## Sémantické sítě matematických znalostí

*Ondřej Šodek*

Katedra aplikované matematiky

Vedoucí práce: Ing. Tomáš Kalvoda, Ph.D.

17. září 2020



---

## Poděkování

Rád bych poděkoval svému vedoucímu práce Ing. Tomáši Kalvodovi, Ph.D. za cenné rady a nesmírnou vstřícnost. Poděkování patří také mé rodině, mým kolegům a mé přítelkyni.



---

# Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o etické přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 46 odst. 6 tohoto zákona tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou, a veškeré jejich dokumentace (dále souhrnně jen „Dílo“), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla, a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu), licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 17. září 2020

.....

České vysoké učení technické v Praze  
Fakulta informačních technologií

© 2020 Ondřej Šodek. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí a nad rámec oprávnění uvedených v Prohlášení na předchozí straně, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Šodek, Ondřej. *Sémantické sítě matematických znalostí*. Bakalářská práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2020.



---

# Abstrakt

Tato bakalářská práce se zabývá zpracováním matematických pojmů do podoby sémantické sítě a využití těchto vztahů k obohacení výsledné vizualizace těchto matematických pojmů. Za tímto účelem byl vytvořen program zpracovávající matematická data ve specifickém formátu, extrahuje sémantické vztahy mezi pojmy a následně vytváří výstup ve webovém rozhraní. Tato vizualizace má za cíl pomoci studentům lépe pochopit vztahy mezi matematickými pojmy.

**Klíčová slova** sémantická síť, znalostní graf, matematická znalost, matematický pojem

---

# Abstract

This thesis is about processing semantic relations from mathematical data and using those relations to enrich the data in a resulting visualisation of those mathematical concepts. For this, a program was created to process mathematical data in a specific format, extract the semantic relations and create a resulting visualisation in a web format. This visualisation is intended for students to better understand the relations between the mathematical concepts.

**Keywords** semantic network, knowledge graph, mathematical knowledge, mathematical concept

---

# Obsah

Úvod	1
<b>1 Cíl práce</b>	<b>3</b>
<b>2 Teorie</b>	<b>5</b>
2.1 Metody pro sémantickou anotaci . . . . .	5
2.2 Zobrazení sémantické sítě . . . . .	7
2.3 Automatická extrakce vztahů . . . . .	7
<b>3 Realizace</b>	<b>9</b>
3.1 Struktura sémantické sítě . . . . .	10
3.2 Formát ukládání dat . . . . .	20
3.3 Program na zpracování a převod dat . . . . .	22
3.4 Zobrazení . . . . .	23
3.5 Dostupnost pro studenty . . . . .	24
<b>4 Výsledky práce</b>	<b>25</b>
<b>Závěr</b>	<b>27</b>
<b>Literatura</b>	<b>29</b>
<b>A Definice pojmů</b>	<b>31</b>
A.1 Seznam použitých zkratk . . . . .	32
<b>B Obsah příloženého CD</b>	<b>33</b>



---

# Úvod

Téměř na každé škole se setkáváme se skutečností, že studenti mají rozdílný způsob a tempo učení. Někteří pochopí látku z přednášky, jiní se látku snaží porozumět později při čtení a počítání příkladů, další se učí pouze z materiálů. Každému funguje něco jiného a v ideálním případě pak všichni látku dříve či později pochopí. Toto však ne vždy nastane.

Jedním z hlavních problémů je nepochopení souvislostí. Ať už studentovi chybí nějaký základ, nebo části látku špatně porozuměl, student si nespojí důležité informace a poté se snaží naučit látku odděleně od toho, co už zná. V horším případě začne být demotivován, protože i přes jeho snahu látku intuitivně nechápe, zatímco ostatní s látkou nemusí mít sebemenší potíže. Dlouhodobě tento problém vede k fragmentaci znalostí, což také přispívá ke špatnému zapamatování látky.

Velkou překážkou v takové situaci je, že student nemůže vědět, kolik toho ve skutečnosti neví. I když se jedná pouze o drobnost, která studentovi unikla, student nepozná, zda by mu k pochopení stačila jedna informace, nebo si musí zopakovat celou kapitolu. Toto se netýká pouze matematických předmětů, ale právě v těchto předmětech to je jeden z největších problémů. V matematice jsou všechny pojmy velmi provázané a téměř každé téma vyžaduje nějaké předchozí znalosti.

Představme si, že má student k dispozici přehled souvislostí všech pojmů s ostatními. Může pak jednoduše najít pro něj problémový pojem, zjistit, které jeho souvislosti zcela neovládá, a z toho rychle poznat, co si musí zopakovat. Nyní do této představy přidejme všechny definice, věty a důsledky. Takovou aplikaci lze potom použít rovnou k vyhledávání chybějících znalostí nebo jako přehled všech důležitých pojmů k opakování. Pokud také propojíme pojmy v aplikaci s materiály předmětu (např. pomocí hypertextových odkazů do webových materiálů předmětu), je možné aplikaci používat jako pomůcku pro vyhledávání v těchto materiálech. Toto propojení je výhodné i pro lepší vysvětlení jednotlivých pojmů, protože materiály předmětu často vysvětlují

jednotlivá tvrzení v mnohem větších detailech a poskytují relevantní příklady.

Jako příklad si představme studenta, který nerozumí, jak se počítá určitý typ příkladu. V aplikaci vyhledá pojem použitý v příkladu a následně vidí všechny relevantní věty spolu s pojmy, se kterými pracují. Student přejde na větu pracující s pojmy vyskytujícími se v příkladu. Zde najde všechny platné vztahy spolu s předpoklady a důsledky platnosti této věty. V důsledcích najde student metodu, jak řešit příklady tohoto typu.

Právě takovou aplikací se tato práce zabývá. Jedná se o sémantickou síť mezi matematickými pojmy, která kromě vztahů mezi pojmy zachycuje i vztahy použité v jednotlivých matematických tvrzeních (definice, věty, důsledky).

Protože matematická oblast je velmi široká a pokrýt ji pomocí sémantické sítě by bylo velmi náročné, tato práce se zaměřuje pouze na rozsah látky z předmětu BI-ZMA (bakalářský předmět základy matematické analýzy) na FIT ČVUT (Fakulta informačních technologií na Českém vysokém učení technickém v Praze).

---

## Cíl práce

Cílem této práce je vytvořit alternativní způsob zobrazení znalostí z předmětu BI-ZMA na FIT ČVUT. Toto alternativní zobrazení má za cíl omezit informace o jednotlivých pojmech a naopak se zaměřuje na pochopení celku (vztahů mezi jednotlivými pojmy). Práce je rozdělena na část teoretickou a praktickou.

Cílem teoretické části práce je prozkoumat teorii sémantických sítí a najít možnosti jejich použití pro reprezentaci znalostí z předmětu BI-ZMA. Bude nalezen způsob reprezentace důležitých pojmů, vlastností, konceptů a vztahů mezi nimi. Bude nalezen vhodný formát zobrazení, který umožní tyto informace jednoduše procházet. Budou prozkoumány vhodné metody automatické extrakce matematických vztahů z textu, které mohou pomoci při tvorbě báze znalostí.

Cílem praktické části práce je použít dostupné materiály z předmětu BI-ZMA a vytvořit z nich zobrazení za použití sémantické sítě. Bude vytvořena sémantická síť důležitých pojmů, vlastností, konceptů a vztahů mezi nimi. Bude vytvořen převodník ze sémantické sítě do finální podoby, jehož součástí budou také odkazy do webových materiálů předmětu BI-ZMA.





---

## Teorie

V této kapitole je popsána teorie sémantických sítí, jejich použití v matematické oblasti a metody potřebné pro realizaci této práce.

Sémantické sítě jsou jednoduchý a srozumitelný nástroj jak reprezentovat znalosti. Jejich použití je zaznamenáno již ve třetím století našeho letopočtu řeckým filosofem Porfyriem [1]. Sémantické sítě nemají obecně definovaný formát, všechny jsou však tvořeny z pojmů určité oblasti a vztahů mezi těmito pojmy.

První zmínka o počítačovém použití sémantická sítě je z roku 1956, kdy Richard Hook Richens navrhnul využití konceptu sémantické sítě jako převodního jazyka pro potřeby strojového překladu textu do jiných jazyků [1]. Sémantické sítě se později začaly rozšiřovat i do oblasti strojového zpracování znalostí. V osmdesátých letech dvacátého století začaly dvě nizozemské univerzity (Groningen a Twente) projekt s názvem *Knowledge Graphs*. Cílem jejich projektu bylo reprezentovat znalosti pro potřeby expertních systémů [2]. Jimi používaná reprezentace znalostí byla nazývána znalostní graf (angl. *knowledge graph*) a jednalo o sémantickou síť s omezeným počtem vztahů mezi jejími pojmy. Postupem času začaly být termíny znalostní graf a sémantická síť chápány jako synonyma.

### 2.1 Metody pro sémantickou anotaci

Ačkoliv metody pro sémantickou anotaci matematických dat existují, je jejich použití v této oblasti omezené. Jedním z důvodů je skutečnost, že experti v matematických oblastech obvykle nemají znalosti sémantického zpracování dat [3]. Dalším důvodem je vysoká náročnost (a nízká návratnost) zápisu matematických dat pomocí existujících sémantických metod [4]. Některé metody (např. MathML, OpenMath) zachycují pouze sémantickou strukturu dat (např.  $x$  je proměnná). Takové metody potom nedokáží zachytit vztahy mezi daty (např. tvrzení  $Z$  je důsledkem věty  $Y$ ). Jiné metody (např. RDF, OWL)

dokáží zachytit obecnější vztahy, jsou však, vzhledem k jejich netriviálnímu formátu, vhodné spíše pro strojovou anotaci dat. Pro potřeby tohoto projektu je třeba nejen sémanticky chápat matematický zápis (pro extrakci pojmů, které jsou v zápisu použité), ale i vztahy mezi jednotlivými pojmy.

Existují i další snahy o sémantickou reprezentaci matematických dat. Jednou z nich je projekt iniciovaný společností Wolfram Research vytvořit sémantickou reprezentaci libovolných matematických tvrzení. Jejich cílem je vytvořit sémantickou reprezentaci přibližně 100 milionů vědeckých prací z matematické oblasti [5]. Jejich projekt by ve své finální podobě jistě dokázal reprezentovat pojmy a vztahy z předmětu BI-ZMA, problémem je rozdílné využití takové sémantické reprezentace. Tato práce se zaměřuje na výuku matematiky a snaží se tedy pomoci studentům pochopit vztahy mezi pojmy. Projekt společnosti Wolfram Research se zabývá strojovým zpracováním (a využitím) těchto vztahů. Zpracování dat pro potřeby počítačového využití je často velmi odlišné od zpracování dat pro výukové účely. Je také nutné použít stejné definice, věty a důsledky jako jsou použité v předmětu BI-ZMA. Definice některých matematických pojmů se mohou mezi autory lišit, což může znamenat potřebu některé definice předefinovat. Taková situace může realizaci velmi ztížit.

Pro potřeby tohoto projektu je nutné provést následující:

- Vytvořit způsob reprezentace pojmů a vztahů mezi pojmy z předmětu BI-ZMA.
- Vytvořit způsob reprezentace matematických tvrzení.
- Vytvořit způsob ukládání těchto dat.
- Extrahovat platné vztahy z materiálů předmětu BI-ZMA.
- Vytvořit výstupní formát pro zobrazování těchto dat.
- Vytvořit program na načítání dat z úložiště, jejich zpracování a převod do výstupního formátu.

Takové použití sémantické reprezentace dat je velmi konkrétně zaměřené a žádný pokus o podobnou realizaci nebyl autorem této práce v době jejího psaní nalezen.

Každá sémantická síť je navržena pro konkrétní využití, neexistuje tedy metoda, která by popisovala způsob vytvoření sémantické sítě pro libovolný účel. Existující metody k tvorbě sémantických sítí jsou buď velmi obecné, nebo zaměřené na velmi úzkou oblast. Jednou z takových oblastí je zpracování přirozeného jazyka, kde je použití sémantických sítí typické. Použití sémantických sítí pro účely zpracovávání znalostí je také časté, jejich použití pro zpracování matematických znalostí už tak časté není. Společnosti, které se tímto problémem zabývají, tyto metody také nezveřejňují. V době psaní

práce nebyly nalezeny žádné metodiky k návrhu sémantické sítě pro potřeby reprezentace matematických znalostí.

Jednou z obecných metod návrhu libovolné sémantické sítě je používat co nejmenší počet vztahů, které umožní splnit požadavky na danou síť [2]. Toto eliminuje případy, kdy je jedna informace zakomponována pomocí více vztahů (případně i více pojmů), které jsou však samostatně bez užitku.

## 2.2 Zobrazení sémantické sítě

Jakmile sémantická síť přesáhne určitou velikost, začne být její graf nepřehledný a grafická vizualizace je tedy nepraktická. Jedním z řešení je vytvořit grafické znázornění pouze vybraných pojmů ze sítě. Jiným řešením je zobrazit pouze jeden pojem ze sítě a k němu zobrazit strukturovaná data založená na vztazích tohoto pojmu s ostatními. Příkladem tohoto postupu je nástroj *Knowledge Graph* od společnosti Google [6], který zobrazuje strukturované informace k právě vyhledanému pojmu. Tato služba je podobná konceptu sémantického webu, společnost Google však neříká, jaké zdroje a metody tato služba využívá. Podobný princip používá také online služba *Wolfram Alpha*, pro kterou byla vytvořena vlastní znalostní báze s vlastní ontologií [7].

Očekává se, že sémantická síť vytvořená v této práci bude značně rozsáhlá a její grafická vizualizace nebude prakticky využitelná. Grafická vizualizace pouze vybraných pojmů ze sítě může najít nějaké využití, měla by však, z důvodu chybějících informací, pouze omezený význam. Tato práce se proto zabývá převážně metodou druhou, a to zobrazovat v jednu chvíli pouze jeden pojem ze sítě, který je doplněn o strukturovaná data vytvořená z jeho vztahů. Jako výstupní formát se jeví vhodné použít webovou HTML stránku. Webové stránky jsou velmi jednoduchý způsob zobrazení informací pro koncové uživatele, zároveň jsou pro tento účel dostatečné a je možné je v případě potřeby jednoduše upravit. Metoda zobrazení je tedy velmi podobná té od společnost Google či Wolfram Research popsaná výše.

## 2.3 Automatická extrakce vztahů

Automatická (strojová) extrakce vztahů z matematického textu je problematická. Jedním problémem je častá kombinace intuitivního a symbolického popisu (např. pro každé okolí  $H_c$  bodu  $c$  existuje okolí  $H_a$  bodu  $a$ ). Pojmy musí být extrahovány z obou typů popisu a často se prolínají. Pro porozumění takovému popisu je nutné rozumět oběma typům popisu a správně je zkombinovat. Dalším problémem je chybovost nástrojů na porozumění a převod textu. Výskyt libovolných chyb velmi snižuje důvěru libovolné znalostní aplikace u koncových uživatelů. Každý strojově vytvořený vztah je tedy potřeba zkontrolovat, což znamená projít a pochopit původní text a ten srovnat s au-

## 2. TEORIE

---

tomaticky vytvořeným vztahem. Toto je srovnatelně složité s tím vytvořit všechny vztahy manuálně.

Vzhledem k neexistenci nástroje na automatickou extrakci matematických vztahů z textu by bylo nutné nejdříve takový nástroj vytvořit a poté provést kontrolu jím vytvořených vztahů popsanou výše. Takový postup nepřináší téměř žádný užitek oproti manuálnímu vytvoření vztahů z materiálů, proto není výhodné se takovým nástrojem dále zabývat.

---

## Realizace

V této kapitole jsou nejdříve krátce popsány části řešení, v dalších sekcích jsou jednotlivé části rozepsány ve větších detailech.

Každý koncept v předmětu BI-ZMA je reprezentován jedním pojmem v síti. Pojmy jsou spojeny dědičným vztahem říkájící, že pojem je specializací (či instancí) jiného pojmu (např. pojem *zobrazení* je specializací pojmu *binární relace*). Dědičný vztah tvoří přirozenou hierarchii matematických pojmů, ale soubor všech pojmů z předmětu BI-ZMA netvoří souvislou strukturu. Aby bylo možné všechny pojmy reprezentovat v jedné struktuře, je nutné tyto oddělené stromy spojit nějakým společným předkem. I po přidání obecnějších pojmů z matematických oblastí mimo předmět BI-ZMA tato struktura není souvislá, proto byl přidán virtuální pojem *objekt*, který je předkem všech pojmů v síti. Existencí společného předka vznikne souvislá hierarchie všech pojmů.

Každý pojem uchovává o svém konceptu informace, které jsou pro předmět důležité. Jedná se o informace: název, alternativní názvy, je specializací, je instancí, definice, platná tvrzení, symbolický zápis a poznámky. Tyto informace jsou podrobně vysvětleny v sekci 3.1.7 Informace o pojmu.

Pro uchovávání těchto dat byl vytvořen jednoduchý textový formát (inspirovaný programovacím jazykem Python), který ukládá data do stromové struktury v závislosti na odsazení řádku. V této struktuře jsou uloženy všechny pojmy a informace o těchto pojmech. Formát jednotlivých informací je závislý na typu dané informace. Formáty každého typu informace je popsán v sekci 3.2 Formát ukládání dat.

Data z datových souborů jsou načteny a strojově zpracovány, což poskytne nové informace o pojmech v síti (např. jaké pojmy jsou použity v matematické větě, tedy na jaké pojmy se daná věta váže). Nově získané informace jsou použity na propojení pojmů dalšími vztahy a také na doplnění informací k jednotlivým pojmům. Tyto informace jsou převedeny do výstupního formátu webové stránky s použitím HTML šablony a šablonovacího systému Jinja2.

### 3.1 Struktura sémantické sítě

Sémantická síť sestává z pojmů určité oblasti a vztahů mezi nimi. Pojmy jsou zadané určenou oblastí a nelze je měnit. Je možné přidat další pojmy, tyto pojmy ale musí dávat smysl v kontextu dané sémantické sítě (může se jednat např. o kombinaci jiných pojmů pro zjednodušení zápisu, je však nutné tyto případy dobře rozmyslet, aby nevznikaly redundance). Hlavní volnost při návrhu sémantické sítě spočívá ve výběru vztahů mezi pojmy této sítě. Struktura sítě (jakým způsobem jsou informace v síti reprezentovány) závisí na použitých vztazích.

Vhodným postupem výběru vztahů je nejdříve určit všechna využití této sémantické sítě a najít všechna využití, která jsou vhodná reprezentovat nějakým propojením pojmů. Jedním takovým vztahem je velmi často používaný dědičný vztah. Někdy ale není vhodné informace reprezentovat vztahem mezi pojmy, typickým příkladem jsou poznámky k pojmu. Poznámky obsahují obecné informace k pojmu a tyto informace není nutné reprezentovat vztahy s jinými pojmy (často to není ani v dané síti možné). Takové informace je lepší uchovávat jako interní data u daného pojmu. Vztahy mezi pojmy by měly reprezentovat nějakou důležitou informaci, se kterou se v projektu pracuje dále.

Obvláště v matematice nastane i situace, kdy je k reprezentaci informací nutné použít výrokovou logiku. Pro tento účel existují další typy sítí [1], které takové případy řeší. V sémantické síti je také možné používat proměnné [1] a také konkrétní hodnoty (konstanty). Takové pojmy mohou mít stejné vztahy jako ostatní pojmy, záleží však na návrhu konkrétní sítě, jak tyto pojmy zpracovává.

#### 3.1.1 Pojmy v síti

K definici vztahů mezi pojmy sémantické sítě je nutné nejdříve definovat, jaké pojmy se v síti budou vyskytovat. Pro potřeby tohoto projektu je nutné v síti reprezentovat všechny důležité pojmy a koncepty z předmětu BI-ZMA. Tyto zahrnují:

- matematické koncepty (např. *zobrazení, relace*)
- matematické věty a důsledky matematických vět (např. *Heineho věta, nutná podmínka konvergence*)

Matematické koncepty jsou jedna z hlavních částí použité sémantické sítě, která má za cíl reprezentovat vztahy mezi těmito pojmy.

Matematické věty reprezentují tvrzení o matematických pojmech. Takové tvrzení může obsahovat informace o mnoha pojmech, je proto vhodné reprezentovat matematickou větu jako pojem v síti, který obsahuje dané tvrzení, a tuto větu propojit s relevantními pojmy.

Důsledek matematické věty je tvrzení, které z této věty vyplývá. Tvrzení reprezentované důsledkem musí být odděleno od tvrzení jeho věty, aby nebylo možné zaměnit důsledek věty s větou samotnou. Proto je vhodné reprezentovat důsledky také jako samostatné pojmy v síti, které jsou spojeny s danou větou.

Sémantická síť tedy musí obsahovat alespoň pojmy reprezentující matematické koncepty, matematické věty a důsledky.

V prvotních verzích tohoto projektu byla snaha také o zakomponování pojmů výrokové logiky spolu s pojmy předmětu a reprezentovat matematické tvrzení jako vztahy mezi pojmy z výrokové logiky a dalšími pojmy. Tato metoda se neukázala jako úspěšná a v dalších verzích byla odstraněna (viz sekce 3.1.4 Parametrizace pojmů).

### 3.1.2 Případy využití

Vztahy použité v sémantické reprezentaci dat jsou velmi závislé na konkrétním využití projektu. Je možné se inspirovat vztahy použitými v jiných projektech, ve výsledku je ale nutné projít všechny případy využití projektu, ze kterých je možné zjistit, jaké vztahy je vhodné v síti použít. Pro potřeby tohoto projektu je nutné reprezentovat následující informace:

- název pojmu a jeho alternativní názvy
- dědičnost pojmu (a zda se jedná o instanci)
- definice pojmu, platná tvrzení týkající se pojmu
- symbolické zápisy pojmu
- případné poznámky k pojmu

Dědičnost pojmů je důležitá pro pochopení hierarchie mezi pojmy. Je to jeden z hlavních způsobů, jak organizovat informace. Dědičnost umožňuje abstrahovat jednotlivé informace, což zjednodušuje učení nových konceptů a pochopení, co daný pojem představuje. Pro pochopení nějakého nového pojmu je mnohem výhodnější znát jeho předka a zaměřit se pouze na to, v čem se tento nový pojem specializuje, tedy v čem se od svého předka liší. Všechny ostatní vlastnosti jsou stejné jako u jeho předka, není tedy důvod se je učit znovu. Matematické pojmy jsou zpravidla specializace (potomci) jiného pojmu a tím tvoří přirozenou hierarchii (např. *zobrazení* je speciální případ *binární relace*, *operace* je speciální případ *zobrazení*, *interval* je speciální případ *množiny*, *uzavřený interval* je speciální případ *intervalu*). Specializace pojmu reprezentuje dědičný vztah a je vhodné ji reprezentovat vztahem mezi pojmy.

Reprezentace definice pojmu je dalším důležitým krokem v tomto projektu. Kombinací dědičnosti a definice pojmu už lze získat poměrně ucelený obrázek o všech matematických pojmech v sémantické síti. Definice pojmu využívá

matematické tvrzení k určení předpokladů pro existenci daného pojmu a případně i vlastností tohoto pojmu. Tato tvrzení používají ostatní pojmy v síti a je z nich možné zjistit, jaké pojmy jsou použity v definici daného pojmu.

Matematické věty jsou nedílnou součástí každé matematické oblasti. Obsahují předpoklad platnosti a tvrzení o jednotlivých pojmech za podmínky, že je předpoklad splněn. Pro potřeby propojení těchto vět s relevantními pojmy, je nutné vědět, jaké pojmy jsou ve větě použity. Důsledek věty je také vhodné propojit s jednotlivými pojmy, které důsledek používá, aby bylo možné podle daného pojmu najít (kromě věty) také konkrétní důsledek.

Alternativní (synonymní) názvy pojmu jsou v projektu použity pro jednodušší vyhledávání pojmů v sémantické síti (a také pro jednodušší zápis dat). Existuje více důvodů, proč mají pojmy synonymní názvy. Jako příklad je možné vzít *interval zleva otevřený a zprava uzavřený*. V tomto názvu je možné přehodit pořadí slov a vznikne podobný název *interval zprava uzavřený a zleva otevřený*. Také je možné název zkrátit na *interval zprava uzavřený* nebo *interval zleva otevřený*. Jiný důvod může být ten, že pojem je nazýván více názvy, jako např. *univerzum* a *univerzální množina*. V projektu je proto vhodné umožnit použití libovolného synonymního názvu k reprezentaci daného pojmu. Názvy pojmů musí být unikátní v celém projektu, není tedy možné mít jeden název, který reprezentuje více různých pojmů. Příkladem může být název *přirozená čísla*, který u někoho může reprezentovat množinu obsahující nulu, jindy množinu bez nuly. Tento název však musí reprezentovat pouze jeden pojem a podle předmětu BI-ZMA reprezentuje množinu neobsahující nulu (je proto definováno i synonymum *přirozená čísla bez nuly*, pro množinu s nulou existuje pojem *přirozená čísla s nulou*). Alternativní názvy nerepresentují žádný vztah mezi pojmy a je tedy vhodné je reprezentovat jako interní informace o pojmu.

Místo názvu pojmu často použít symbol reprezentující daný pojem, proto je nutné ukládat a zobrazovat i symbolický zápis. Těchto symbolů se může pro daný pojem používat i více. Je možné ve výstupu informativně zobrazit všechny možné způsoby symbolického zápisu, tedy i pro studenty, kteří jsou zvyklí na jiné značení. Symbolické zápisy nerepresentují žádný vztah mezi pojmy a je tedy vhodné je reprezentovat jako interní informace o pojmu.

Poznámky mají za cíl zdůraznit nějakou informaci. Měl by existovat dobrý důvod, proč danou informaci zdůrazňují. Jedním z hlavních důvodů je upozornění na častou chybu, které se studenti dopouštějí. Dalším důvodem je přidání informace, kterou nelze v síti jinak reprezentovat (např. rozšíření znalostí nad rámec předmětu), nebo zvýraznění nějaké důležité informace o pojmu. Poznámky jsou závislé na použití projektu a je možné je využít i jiným způsobem. Poznámky jsou reprezentovány jako interní informace o pojmu, nerepresentují žádný vztah mezi pojmy.



### 3.1.3 Typy pojmů

Protože matematické pojmy tvoří přirozenou hierarchii, mají téměř všechny pojmy nějakého předka. Výjimkou jsou pouze ty nejjobecnější pojmy, které nejsou specializací žádného jiného pojmu. Předmět BI-ZMA obsahuje omezený počet pojmů a neobsahuje předky některých pojmů, které jsou v předmětu definovány. Proto je síť rozšířena o předky těchto pojmů z oblastí mimo předmět BI-ZMA. Většina těchto pojmů (předků) je definována, nebo alespoň zmíněna, v předmětu BI-PKM (přípravný kurz matematiky pro budoucí studenty prvního ročníku).

Nejobecnější pojmy v síti jsou *množina*, *n-tice*, *tvrzení*, *výraz* (ve smyslu konečná posloupnost symbolů). Tyto pojmy nejsou nutně nejjobecnější v matematickém oboru (např. pojem *n-tice* lze definovat pomocí množin), ale pro účely předmětu BI-ZMA není třeba zacházet do takové hloubky či tyto pojmy formálně definovat (více v sekci 3.1.6 Otevřené konce). I pro tyto pojmy je ale vhodné najít nějaký společný typ, aby bylo tímto typem možné specifikovat libovolný pojem ze sítě. Proto byl přidán pojem *objekt*, který je předkem všech pojmů v síti. Každý pojem ze sítě je tedy potomkem pojmu *objekt*. Pojem *objekt* ale není ničím pro výuku zajímavý a není jej proto nutné zobrazovat ve výsledném zobrazení. Byl proto přidán způsob specifikovat skrytý pojem, který se nemá zobrazovat ve výsledném zobrazení. Všechny skryté pojmy působí pouze jako další typy pojmů a nejsou pro uživatele jinak důležité.

Každý matematický koncept může reprezentovat nějakou konkrétní hodnotu (instanci) tohoto konceptu (např. *množina reálných čísel* je konkrétní instancí *číselné množiny*). Instance pojmů jsou obvykle jednoduché na pochopení a není nutné je uvádět, pokud nejsou nikde použity, je lepší se zaměřit na pochopení obecného konceptu, který studentům umožní pochopit i všechny jeho instance. Výjimkou jsou matematické věty a důsledky. Každá věta (např. *Heineho věta*) je instance obecného konceptu *matematická věta*. Instance se od obecných pojmů liší tím, že koncepty mají své instance, instance už žádné další instance nemají. Libovolný matematický koncept reprezentuje všechny své instance. Každý pojem v síti tedy reprezentuje nějaký matematický koncept, nebo jeho instanci.

V tomto textu je dále používán termín *typ pojmu*, který reprezentuje specifikovaný pojem a všechny jeho specializace (tedy jeho přímé i nepřímé potomky). Každý pojem reprezentuje nějaký matematický koncept a všechny jeho instance, typ pojmu tedy také reprezentuje všechny instance specifikovaných pojmů. Například *interval* je typ pojmu, který zahrnuje *interval*, *otevřený interval*, *uzavřený interval* a další pojmy včetně konkrétních instancí jako například *uzavřený interval*  $\langle 1, 2 \rangle$ .

Typy pojmů jsou důležité jak pro lepší představu o pojmech, tak pro matematická tvrzení, kdy je možné definovat proměnné nějakého typu a tím omezit, jaký pojem (případně jakou hodnotu) může tato proměnná reprezentovat. Omezovat typy je zapotřebí, když je cílem podat co nejvíce informací

o tom, s čím daný pojem pracuje. Např. operace *sčítání čísel* pracuje se dvěma objekty typu *číslo* (tedy pracuje se všemi specializacemi pojmu *číslo*, jednou z nich je *komplexní číslo*). Je také možné říci, že prvkem množiny je pojem typu *objekt*, což znamená že typ pojmu není specifikován a může představovat libovolný pojem ze sítě.

### 3.1.4 Parametrizace pojmů

V prvních verzích projektu byla použita tzv. parametrizace pojmů jako způsob, jak jednoduše definovat určitou specializaci pojmu. Díky dědičnosti může jeden typ pojmu reprezentovat všechny jeho potomky (specializace). Pokud je naopak vhodné reprezentovat určitou specializaci tohoto pojmu, je nutné použít pojem, který tuto specializaci reprezentuje (tedy dědí z daného pojmu a v definici má podmínky pro jeho specializaci). Ačkoliv v pojmech předmětu BI-ZMA je pouze omezený počet pojmů a je možné definovat takový pojem jednotlivě, parametrizace přináší způsob, jak řešit jednoduché specializace pojmů (kterých může být i neomezené množství). Tyto specializace umožňují rychle sdělit informace o nějakém pojmu za pomoci nějaké informace - parametru. Parametry vždy reprezentují proměnnou nějakého typu, tato proměnná je použita v definici pojmu (může být použita také u dědičného vztahu) a určuje nějaké omezení tohoto pojmu.

Příkladem tohoto použití je pojem *n-ární relace*, což je jeden pojem reprezentující všechny specializace relace pro libovolné  $n$ . Pokud je nutné specifikovat relaci mezi dvěma množinami, je možné použít pojem *binární relace*, což je *n-ární relace* pro  $n = 2$ . Zde se nabízí zjednodušení ve formě parametrů, které předají pojmu *n-ární relace* informaci o čísle  $n$ . Pojem *n-ární relace* ale také potřebuje znát množiny, mezi kterými tuto relaci definuje. Protože není předem známo, mezi kolika množinami je tato relace definována, je vhodné jako parametr předat  $n$ -tici množin, které tyto množiny reprezentují. Tato  $n$ -tice navíc nese informaci o čísle  $n$  a proto není potřeba, aby pojem *n-ární relace* přijímal jiný parametr než  $n$ -tici množin. Parametry nejsou povinné a je stále možné specifikovat pojem typu *n-ární relace* bez parametrů, který reprezentuje všechny specializace *n-ární relace*. Pojem *binární relace* samozřejmě reprezentuje stejný koncept jako *n-ární relace* pro  $n = 2$ , tyto dva pojmy se ale v síti liší svými parametry a proto je vhodné v síti reprezentovat oba pojmy. Pojem *binární relace* má jako parametry dvě množiny a je to synonymum pojmu *binární relace mezi množinami*. Existuje také pojem *binární relace na množině*, který má pouze jeden parametr (typu *množina*), protože se jedná o binární relace mezi dvěma stejnými množinami. Podobně tomu je pro pojem *n-ární relace mezi množinami*, která má jako parametr  $n$ -tici množin ( $n$ -tice uchovává informaci o čísle  $n$ ), *n-ární relace na množině* má jako parametr číslo  $n$  typu *přirozené číslo s nulou* (přirozené číslo nebo nulu) a druhý parametr *množinu* (jedná se o relace mezi  $n$  stejnými množinami).

Je tedy nutné uchovávat informaci o tom, kolik parametrů (a jakého typu)

pojmem přijímá. U parametrů záleží na pořadí a je nutné si také uchovávat informaci o pořadí jednotlivých parametrů. Každý parametr reprezentuje proměnnou s určitým názvem (např.  $x$ ), která je použita v definici pojmu. Díky tomu také parametry umožňují definovat složitější symbolický zápis, s použitím konkrétních proměnných. Tyto parametry mohou být ve výstupu nahrazeny za konkrétní hodnoty (nebo konkrétní názvy proměnných) a zobrazeny.

Parametrizace však přináší řadu problémů. Jaké parametry by měl pojem mít? Jsou takové parametry dostatečně obecné aby tím šlo specifikovat vše potřebné? Na předchozích příkladech jsou vidět jisté problémy. Někdy je výhodné specifikovat *n-ární relaci* pro určité  $n$  bez specifikování jednotlivých množin, potom je naopak tento pojem příliš obecný a na specifikaci ternární relace pro tři množiny je nutné separátní pojem.

U některých pojmů může použití parametrů změnit typ těchto pojmů. Například *číselný součet* může přijmout dva parametry a potom reprezentuje výsledek této operace typu *číslo*, pokud je však *číselný součet* specifikováno bez parametrů, reprezentuje *binární operaci*. Toto by bylo Proto je nutné do sítě přidat způsob reprezentování informace říkající, zda pojem s parametry mění svůj typ (a na jaký typ se mění).

Ve výsledku, parametry nepřinášejí mnoho užitku a naopak způsobují nárůst pojmů a komplexity dat. Proto není parametry vhodné používat na specifikaci omezeného počtu specializací pojmu, obvykle je lepším řešením specifikovat každý pojem zvlášť.

Parametry jsou však stále výhodné používat v omezené míře pro definování typů proměnných použitých v definici daného pojmu.

### 3.1.5 Reprezentace matematického zápisu

Samotné vztahy mezi matematickými pojmy dokáží reprezentovat nějaké jednoduché informace o těchto pojmech (např. dědičnost/specializace), nedokáží však reprezentovat složitější informace jako matematická tvrzení (použitá mj. v definici pojmu). Matematická tvrzení používají výrokovou logiku, proměnné, konstanty, a pojmy pro reprezentaci vztahů mezi proměnnými. Jsou to tedy pevně strukturované celky informací.

Tato skutečnost znamená, že matematická tvrzení musí být reprezentována jiným způsobem než ostatní informace v síti. Matematické tvrzení je potom možné určitým vztahem spojit s pojmem (a tím toto tvrzení zahrnout do sítě) přes určitý vztah. Tento vztah definuje význam tohoto tvrzení pro daný pojem. Může se jednat o definici pojmu, matematickou větu, důsledek, nebo obecně platný matematický vztah. Návrh sémantické sítě určuje, jak jsou tyto případy rozlišeny a tedy řeší, jak jsou informace z tohoto tvrzení použity.

Pro tento projekt je vhodné získat z matematických tvrzení následující informace:

- předpoklady pro platnost tvrzení

### 3. REALIZACE

---

- důsledek tvrzení
- typy pojmů vyskytující se v předpokladech a v důsledku tvrzení

Tyto informace jsou užitečné podle toho, kde se vyskytují. Předpoklady tvrzení jsou zvláště důležité v definicích, protože obsahují pojmy, které jsou nutné znát pro definici daného pojmu. Může být také výhodné vyzdvihnout předpoklady matematické věty. Důsledek může specifikovat nové platné vztahy či definovat pojem. Pojmy, které se v těchto tvrzeních vyskytují, je tedy možné spojit určitým typem vztahu s daným tvrzením (a tedy i pojmem u kterého se tvrzení nachází). Tento typ vztahu je definován podle typu pojmu a také podle použití pojmu v daném tvrzení.

Jedním ze způsobů, jak extrahovat vztahy z matematických tvrzení, je psát matematická tvrzení ve strojově čitelném formátu. Takový způsob musí umožňovat reprezentovat následující:

- výroková logika (negace, konjunkce, disjunkce, implikace, ...)
- definování proměnné určitého typu
- přiřazení hodnoty nějaké proměnné
- aplikování vztahů mezi proměnnými

Výroková logika je možná řešit vytvořením pojmů v síti reprezentující jednotlivé výrokové formule (např. konjunkce, implikace) a použitím těchto pojmů spolu s proměnnými k sestavení daného tvrzení. Typ pojmu je možné určit dědičným vztahem na pojem do sítě (např.  $A$  je *množina*).

Proměnné reprezentující nějaký pojem nebo přímo jeho konkrétní hodnotu. Určení konkrétní hodnoty proměnné by pomocí vztahů s ostatními pojmy bylo náročné, proto je možné konkrétní hodnotu uložit jako interní informaci u této proměnné.

Kromě výrokové logiky a definování proměnných je také nutné dávat proměnné do vztahů (např.  $a < b$ ), případně určovat vlastnosti (např.  $f \circ g$  je *prostá funkce*), a také provádět operace s proměnnými (např.  $a + b$ ). Tyto vztahy a operace mohou samy být pojmy v sémantické síti, jejich aplikace na proměnné může být řešena parametry těchto pojmů.

Jediným cílem této reprezentace je, aby program zpracovávající tuto reprezentaci dokázal určit potřebné informace vypsané výše a také vypsát toto tvrzení do výstupního formátu. Vhodným řešením je vytvořit stromovou strukturu všech částí daného tvrzení tak, aby zachovala všechny informace původního tvrzení. Například implikace má 2 potomky, první z nich je předpoklad implikace, druhý důsledek. Oba potomci mohou sestávat z více výroků či podstromů (samostatné stromové struktury).

Ačkoliv byl v prvotních verzích projektu takový způsob použit pro extrakci vztahů, není pro tento projekt příliš vhodný. Vzhledem k tomu, že tento projekt má být podpůrným prostředkem pro studenty, je vhodné, aby byl zachován intuitivní zápis tvrzení (textový opis nějaké části tvrzení), který je použit v materiálech předmětu. Takový způsob je mnohdy jednodušěji pochopitelný, než kdyby k tomuto zápisu byl použit symbolický zápis. Pojmy je v omezené míře možné i na textový popis, toto však není vhodné kvůli různým případům použití daného pojmu, skloňování a pořadí slov, apod.

Byla také snaha použít tzv. pomocné pojmy, tedy pojmy, které zahrnují více vztahů mezi sebou (např. existuje limita funkce v bodě), ale toto vedlo k narůstající komplexitě a nepřineslo velký užitek.

V projektu je proto použit velmi podobný zápis tvrzení jako v materiálech předmětu BI-ZMA, a to psanou formou ve formátu L<sup>A</sup>T<sub>E</sub>X. Pojmy z těchto tvrzení jsou vypsány mimo samotné tvrzení, které jsou poté zpracovány a použity v síti.

### 3.1.6 Otevřené konce

Vzhledem k rozsáhlosti matematické analýzy (a matematické domény obecně) je tento projekt omezen pouze na pojmy z předmětu BI-ZMA. V síti je několik pojmů, které jsou definovány pouze intuitivně, protože jejich definice jde mimo rozsah předmětu BI-ZMA. Hlavním příkladem takového pojmu je pojem množina. Množina je v předmětu BI-ZMA chápána pouze jako jednoduchá neuspořádaná kolekce (sada) objektů. Tato naivní definice však vede k paradoxu zvanému Russelův paradox, což je zmíněno i v přípravném kurzu matematiky (BI-PKM). Proto je použit tzv. otevřený konec, pojem tedy není explicitně definován, pokud si chce student rozšířit znalosti nad rámec předmětu BI-ZMA, je odkázán v poznámce daného pojmu.

### 3.1.7 Informace o pojmech

Každý pojem v síti je reprezentován datovou strukturou reprezentující informace o tomto pojmu. Tyto informace zahrnují: název, alternativní názvy, dědičnost a instance, definice, platné vztahy, symbolický zápis, poznámky.

Kromě názvu a dědičnosti nejsou tyto informace povinné. Definice se specifikuje jen u obecných pojmů, u instancí se definice nepoužívá, místo toho jsou použity platné vztahy, které nějakým způsobem doplňují definici obecného pojmu).

#### 3.1.7.1 Název

Jedná se o název, kterým je pojem reprezentován. Všechny názvy pojmů musí být v projektu unikátní.

#### 3.1.7.2 Alternativní názvy

Alternativní názvy určují synonyma daného pojmu. Tato synonyma pojmu musí být v datech unikátní, nesmí se jmenovat stejně jako jiné pojmy nebo jejich alternativní názvy. Pro zadávání dat není důležité, zda je použit název pojmu nebo jeho alternativní název. Pojem a jeho synonyma reprezentují stejný objekt. Každý alternativní název pojmu je řetězec znaků.

**Příklad:**

Alternativní název pojmu *číselná řada* je *řada*.

#### 3.1.7.3 Dědičnost a instance

Dědičnost říká, že pojem je specializací jiného pojmu. Jedná se o dědičný vztah mezi těmito pojmy. V angličtině se tento vztah často nazývá *isa* nebo *is-a*. Dědičný vztah je tranzitivní, každý pojem tedy dědí ze všech svých (přímých i nepřímých) předků. Informace o dědičnosti je u každého pojmu povinná a předkem pojmu musí být nějaký jiný pojem v síti. Jedinou výjimkou je nejobecnější pojem ze sítě: *objekt*. Pokud se jedná o instanci pojmu, je místo vztahu pro dědičnost použit vztah pro instanci pojmu.

**Příklad:**

Pojem *reálná čísla* je specializace pojmu *číselná množina* (je to nepřímý potomek pojmu *množina*).

Pojem *zobrazení* je specializace pojmu *binární relace*.

Pojem *nutná podmínka konvergence* je instance pojmu *matematická věta*.

#### 3.1.7.4 Definice

Definice je specifikována ve formě matematického tvrzení. Instance pojmu definici nemají (např. pro zápis tvrzení u konkrétní věty je nutné použít typ informace **Platné vztahy**).

**Příklad:**

Pojem *binární relace* (s parametry  $X, Y$ , oba typu *množina*) je definován jako podmnožina kartézského součinu  $X, Y$ .

#### 3.1.7.5 Platné vztahy

Platné vztahy definují nějaké matematické tvrzení. Může se jednat o matematickou větu, důsledek, nebo obecně platné tvrzení a odlišují se tím, u jakého typu pojmu se nachází. Pokud je tvrzení u pojmu typu *věta*, pak se jedná o tvrzení této věty. Podobně u pojmu typu *důsledek* se jedná o tvrzení důsledku. V ostatních případech se jedná o platné tvrzení o tomto pojmu.

**Příklad:**

Pro pojem *přirozená čísla* platí, že je vlastní podmnožina pojmu *přirozená čísla s nulou*.

Pro větu *nutná podmínka konvergence* platí implikace:

$$\text{řada } \sum_{k=0}^{\infty} a_k \text{ konverguje} \implies \lim_{k \rightarrow \infty} a_k = 0.$$

### 3.1.7.6 Symbolický zápis

Tato informace určuje symbolický zápis (případně více symbolických zápisů) daného pojmu a slouží pro zobrazení možných symbolických zápisů tohoto pojmu. Každý symbolický zápis pojmu je řetězec znaků a jedná se o reprezentaci symbolu ve formátu  $\text{\LaTeX}$ .

**Příklad:**

symbolický zápis pojmu *přirozená čísla* je  $\text{\mathbb{N}}$  pro  $\text{\LaTeX}$  formát.

### 3.1.7.7 Poznámky

Poznámky reprezentují důležité informace o pojmu, které není možné (nebo není vhodné) reprezentovat jiným způsobem. Každá poznámka je řetězec znaků.

**Příklad:**

pojem *číselná řada* má poznámku „Pozor, neplést s pojmem posloupnost!“

### 3.1.8 Vztahy mezi pojmy

Pro potřeby projektu je nutné zpracovat vztahy mezi pojmy z informací o jednotlivých pojmech a poté tyto vztahy uplatnit ve výsledné vizualizaci. Některé z těchto vztahů jsou triviální, protože jsou specifikovány přímo v datové struktuře daného pojmu. Jiné vyžadují zpracování konkrétních dat (např. zpracování reprezentace pojmů z matematického tvrzení).

Vztahy mezi pojmy zahrnují:

- je specializací
- je instancí
- je předpokladem k pojmu
- je důsledkem
- je vlastností pojmu

Vztahy jsou vždy mezi dvěma pojmy v síti. Tyto pojmy mohou reprezentovat buď obecné matematické koncepty (např. množina, matematická věta) nebo jejich instance (např. Heineho věta). Vztahy jsou vždy jednosměrné.

Kromě výše uvedených vztahů jsou použity také další pomocné vztahy, které spojují pojmy na základě nějakých užitečných informací (např. skládání funkcí je operace s pojmem funkce).

### 3.1.8.1 Je specializací

Určuje předka daného pojmu. Jedná se o dědičný vztah mezi pojmy a říká, že pojem je obecnou specializací daného předka.

### 3.1.8.2 Je instancí

Říká, že daný pojem je instancí jiného pojmu. Tento vztah je použit převážně u konkrétních vět a důsledků, které jsou instancemi pojmu *matematická věta*, resp. *důsledek*. Každý pojem je (až na výjimky) buď specializací nebo instancí jiného pojmu.

### 3.1.8.3 Je předpokladem k pojmu

Vztah, který říká, že znalost pojmu je předpokladem ke správnému pochopení jiného pojmu. Tedy tento pojem je z pravidla použitý v předpokladech u definice cílového pojmu.

### 3.1.8.4 Je důsledkem

Říká, že daný pojem (typu *důsledek*) je důsledkem matematické věty (pojem typu *matematická věta*).

### 3.1.8.5 Je vlastností pojmu

Říká, že daný pojem je vlastnost jiného pojmu.

## 3.2 Formát ukládání dat

Aby bylo možné přidávat nová data nebo data v sémantické síti upravovat, je nejdříve nutné tato data uložit. Formát ukládání dat musí umožňovat reprezentovat všechny typy informací o daném pojmu (viz sekce 3.1.7 Informace o pojmu).

Formát musí být schopen reprezentovat tyto informace:

- pojmy
- informace o pojmech (mohou mít různou strukturu)
  - proměnné
  - odkazy na jiné pojmy v síti (i s parametry)
  - matematické tvrzení (viz sekce 3.1.5 Reprezentace matematického zápisu)



Mezi další požadované vlastnosti formátu patří jednoduchost zápisu a čitelnost.

Dílčím cílem práce je tedy najít nebo vytvořit formát, který umožní reprezentovat všechna potřebná data a bude zároveň co nejjednodušší na zápis a čtení člověkem. Výhodou použití nějakého existujícího formátu je možnost použít již vytvořený parser. Nevýhodou je, že se takový formát nedá rozšířit a musí tedy existovat způsob, jak jej použít pro všechna data projektu. Konkrétní existující formáty už žádné další výhody nepřinesou, mohou však přinést nevýhody v čitelnosti, složitosti zápisu nebo nutnosti řešit nějaký problematický zápis pomocí tohoto formátu. Např. formát JSON je velmi složitý na zápis a čtení, protože je určen na strojovou serializaci dat, není tedy vhodný pro použití v tomto projektu. Jiným příkladem je formát YAML, který dokáže reprezentovat velké množství typů dat, je dobře čitelný a není příliš složitý na zápis. Je tedy vhodným kandidátem pro použití v tomto projektu.

I přes existenci vhodných kandidátů na formát ukládání dat, v počáteční fázi projektu byl zvolen vlastní formát. Hlavní důvod pro tuto volbu byla složitost reprezentace matematického zápisu, kde je největší komplexita zápisu dat. U reprezentace matematického zápisu je žádoucí psát více informací na jeden řádek, což je u existujících formátů možné zpravidla jen pokud se jedná o řetězce znaků (ohraňené uvozovkami). Tato skutečnost znamená zvýšenou složitost zápisu a čtení, navíc by bylo nutné na takové řetězce vytvořit parser. Pro jejich zpracování je ale také potřeba znát kontext sousedních řádků, který parser zpracovávající jednotlivé řetězce nemá k dispozici. Alternativně by bylo možné psát matematická tvrzení jiným způsobem, který daný formát dokáže zpracovávat. Toto by však znamenalo atomicky rozdělit matematický zápis, což implikuje složitější zápis a čitelnost. Proto bylo rozhodnuto vytvořit vlastní formát ukládání dat, kde má parser větší kontrolu nad daty a může využívat kontext okolních dat.

Vlastní formát je oproti existujícím formátům vytvořený na míru a nepoužívá žádné přebytečné znaky (dvojtečky, pomlčky). Hlavním principem tohoto formátu je stromová struktura daná odsazením řádků (inspirovaná jazykem Python). Jednotlivé řádky mohou mít různý formát v závislosti na jejich odsazení v souboru a také v závislosti na jejich předkovi ve stromové struktuře. Tento formát se velmi osvědčil a byl postupně rozšiřován až do aktuální verze projektu.

Všechny soubory s daty musí být ve složce *data* nebo v libovolné složce nacházející se pod složkou *data*. Názvy těchto souborů nesmí začínat tečkou, jinak budou ignorovány. Soubory musí být v kódování *UTF-8* a musí být odsazeny buď tabulátory nebo mezerami, nesmí však tyto způsoby odsazování v rámci jednoho souboru kombinovat. Komentáře v souboru jsou uvozovány symbolem mřížka (**#**) a mohou se vyskytovat kdekoli na řádce.

Všechny pojmy v datech mají názvy bez mezer. Pokud je název pojmu vytvořen z více slov, je nutné nahradit mezery za znak podtržení (**\_**).

Pojmy sémantické sítě jsou na řádcích bez odsazení (reprezentují název

tohoto pojmu). Jeho přímí potomci ve stromové struktuře (řádky s jedním odsazením) reprezentují typ informace o pojmu (např. `alternativní_názvy`). Potomci řádku s jedním odsazením (tedy řádky se dvěma nebo více odsazeními) reprezentují samotné informace daného typu. Struktura těchto dat je závislá na typu těchto informací.

Samotnou definici datové struktury pojmů je možné nalézt na přiloženém CD v souboru `README.md`.

## 3.3 Program na zpracování a převod dat

Cílem programu je vytvořit zobrazení z dat sémantické sítě. Program má k dispozici uložená data ve specifikovaném formátu a má za úkol vytvořit z těchto dat požadované zobrazení.

Aby bylo možné data převést do výsledného zobrazení, je nutné tato data nejdříve předzpracovat a interně propojit, aby bylo možné tato propojení převést na hypertextové odkazy mezi jednotlivými pojmy ve výsledném zobrazení.

Dalším důvodem předzpracování dat jsou různé typy informací, které očekávají různý formát dat. Může se jednat o jednoduché informace, které pouze specifikují název pojmu ze sítě (např. dědičnost pojmu). Ale zpracování reprezentace matematického zápisu. Matematický zápis je v datech reprezentován tak, aby jej bylo možné strojově načíst a bylo z něj možné zjistit potřebné informace. Tento zápis je také nutné transformovat do formátu, který je lidsky pochopitelný a dá se zobrazit ve výsledném zobrazení. Každý typ informace tedy stanovuje, jak mají být tato informace zpracována.

Předzpracování také doplní data na chybějících místech. Metoda zápisu dat se řídí pravidlem *DRY*, což znamená neopakuj se (angl. *don't repeat yourself*). V datech je tedy informace uložena pouze jednou, ačkoliv může být potřeba na více místech. Předzpracování dat tuto informaci doplní na všechna relevantní místa. Toto je použito mj. u dědičného vztahu, kde pojem dědí ze všech jeho předků a tím dědí také všechny vlastnosti těchto předků. Při zobrazení je tedy vhodné napsat přímo k tomuto pojmu všechny vlastnosti i spolu s těmi, které zdědil, aby uživatel nemusel procházet všechny předky pojmu ke zjištění jeho vlastností.

### 3.3.1 Postup programu

Program nejdříve najde všechny soubory nacházející se kdekoli pod složkou *data* (tedy i v libovolné podložce) a vyřadí z nich všechny soubory jejichž název začíná tečkou (v kontextu Unixových systémů se jedná o skryté soubory). Poté začne postupně zpracovávat jednotlivé soubory.

Každý soubor je převeden do stromové struktury, kde pro soubor je vytvořen kořenový uzel. Každý řádek bez odsazení je přímý potomek tohoto kořene, každý řádek s nenulovým odsazením je přímým potomkem předchozího řádku s menším odsazením. Soubor musí začínat řádkem bez odsazení, jinak se jedná

o řádek bez předka a tento řádek je ignorován. Libovolné komentáře jsou odstraněny a prázdné řádky jsou ignorovány. Všechny tyto stromy jsou zasazeny do globálního stromu, který působí jako datová struktura obsahující všechna data z datových souborů.

Tato stromová struktura je dále zpracovávána a jsou z ní extrahovány všechny informace o pojmech. Zpracování dat probíhá v určitém pořadí. Toto zabraňuje situacím, kdy je potřeba znát informaci, která ještě není načtená. Nejdříve jsou načteny a uloženy všechny názvy pojmů spolu se všemi jejich alternativními názvy. Všechny tyto názvy jsou kontrolovány pro jejich unikátnost v rámci sémantické sítě a v případě nalezení kolize je poslední název ignorován a vypsána chybová hláška. Jakmile jsou zpracovány všechny názvy pojmů v sémantické síti, jsou zpracována všechna data o pojmech, kromě těch reprezentující matematický zápis. Pro zpracování matematického zápisu je nutné znát všechny alternativní názvy pojmů, dědičné vztahy, parametry, výstupní typ a symbolické zápisy. Jsou proto zpracovávány jako poslední.

### 3.4 Zobrazení

K využití tohoto projektu pro účely výuky je nutné data ze sémantické sítě zobrazit v jednoduše dostupném a použitelném formátu, který umožní studentům jednoduše najít informace, které potřebují. Formát zobrazení byl v kapitole teorie (2) rozebrán a byl vybrán způsob zobrazení pomocí webové stránky, která zobrazuje pojmy jako jednotlivé bloky se strukturovanými daty ze sítě.

U každého pojmu je nutné zobrazit následující informace:

- název
- alternativní názvy
- symbolický zápis
- dědičnost/instance
- definice
- platná tvrzení
- poznámky
- Odkazy na souvislé pojmy (např. věty a důsledky, vlastnosti, apod.)

Výše uvedené informace jsou obohaceny o informace zpracované ze vztahů daného pojmu s ostatními pojmy.

Každý druh informace je vhodné strukturovat do vizuálně odděleného bloku a přehledně označit, aby bylo od pohledu zřejmé, jaké informace tento blok obsahuje.

Podobný způsob zobrazení informací o pojmu používá WolframAlpha, který stejně jako tento projekt obsahuje bloky alternativní názvy a definice. WolframAlpha také obsahuje informace o klasifikaci daného pojmu, ve které zobrazuje témata z matematické oblasti, ve kterých se pojem používá. Mimo klasifikaci používá také *MSC2010* [8] identifikaci pojmů. Taková klasifikace/identifikace v tomto projektu nemá velké využití, protože tento projekt má za cíl zlepšit pochopení pojmů z předmětu BI-ZMA, nikoliv obecné rozšíření znalostí. Do budoucna by však mohlo být využití takové klasifikace/identifikace výhodné, zvláště u hraničních pojmů oblasti matematické analýzy.

Pro vyhledávání v pojmech sémantické sítě je zobrazení doplněno o abecedně seřazený klikatelný obsah, ve kterém jsou všechny názvy a synonyma pojmů ze sítě. Tento obsah je dále doplněn o interaktivní filtr, který pro zadaný vstup zobrazí pouze relevantní pojmy.

Odkazy do materiálů předmětu jsou řešeny klikatelnou ikonou u každého pojmu, která uživateli otevře webovou verzi studijních materiálů na daném pojmu. U některých pojmů je k dispozici i odkaz na vzorový příklad, který má za cíl pojem studentům ještě více přiblížit.

## 3.5 Dostupnost pro studenty

Výsledné zobrazení projektu je ve formě webové stránky, je tedy možné takovou stránku jednoduše distribuovat mezi studenty předmětu BI-ZMA. Toto je výborná vlastnost pro používání této aplikace offline, nevýhoda je, že po změně dat v projektu bude tato stránka zastaralá. Lepším řešením je centrální způsob správy a vytváření takového zobrazení pomocí *GitLab CI/CD*. Tento způsob je převzatý ze způsobu správy materiálů předmětu BI-ZMA.

Myšlenka *CI/CD* spočívá v nastavení *Git* repozitáře, aby při dostupnosti nové verze dat (nebo programu) tato data nově zpracoval a vytvořil z nich všechny výstupní soubory, které jsou potřebné. Tento proces je možné nastavit takovým způsobem, aby nově vytvořené soubory (výstup programu) nahradil předchozí verzi dostupnou na nějaké známé webové adrese. Tímto se zaručí, že bude dostupná vždy nejnovější verze výstupu.

---

## Výsledky práce

Výsledkem této práce je aplikace poskytující alternativní zobrazení pojmů z předmětu BI-ZMA. Tato aplikace je ve webové podobě a je volně dostupná všem studentům tohoto předmětu. Má za cíl pomoci studentům pochopit vztahy mezi (velmi provázanými) pojmy z tohoto předmětu, který má dlouhodobě jednu z nejnižších průchodností ze všech předmětů určených pro studenty prvního semestru FIT ČVUT.

Dále je v této práci vytvořen formát pro jednoduchý zápis matematických pojmů, jejich vztahů a matematických tvrzení. To mj. umožňuje jednoduché definování nových pojmů, které mohou být využity v matematických tvrzeních. Díky tomu jsou data v tomto projektu jednoduše upravitelná a rozšiřitelná.

Data projektu je možné upravit či rozšířit pro potřeby jiných oblastí (např. jiných předmětů), kde výstupem je jednoduché zobrazení pojmů z libovolné oblasti s informacemi o jejich hierarchii a jejich definicích.



---

## Závěr

Cílem této práce bylo vytvořit alternativní způsob zobrazení znalostí z předmětu BI-ZMA.

Tento cíl se podařilo naplnit a výsledné zobrazení je velmi provázané a přehledné zobrazení matematických pojmů.

Projekt je možné upravit (či rozšířit) pro data z jiné matematické oblasti. Pro tento účel je nutné vytvořit datové soubory ve formátu specifikovaném v kapitole. Data je doporučeno dělit do relevantních souborů, tak aby jednotlivé soubory nebyly příliš velké. Pokud je cílem zahrnout data z více různých oblastí (např. předmětů), je vhodné tyto soubory rozmístit do jednotlivých složek (případně podložek) podle jejich oblasti.

Je také možné upravit výsledné zobrazení tohoto projektu. Jednou z možností je upravit nebo vytvořit novou HTML šablonu a případně upravit Jinja2 konstrukce, které do šablony doplňují obsah z datových souborů. Druhou možností je upravit data, která jsou předávána jako kontext šabloně. Tento kontext je možné v programu upravit a tím šabloně předat upravená data podle potřeby. Třetí možností je nepoužívat šablonovací systém a vyexportovat data v potřebném formátu (např. JSON), který je následně importován a zpracován jinou aplikací.





---

## Literatura

- [1] Sowa, J. F.: Semantic Networks. [online], [cit. 2019-04-22]. Dostupné z: <http://www.jfsowa.com/pubs/semnet.pdf>
- [2] Nurdyati, S.; Hoede, C.: 25 YEARS DEVELOPMENT OF KNOWLEDGE GRAPH THEORY: THE RESULTS AND THE CHALLENGE. 2008, ISSN 1874-4850. Dostupné z: <https://core.ac.uk/download/pdf/11468596.pdf>
- [3] Lange, C.: Ontologies and Languages for Representing Mathematical Knowledge on the Semantic Web. *Semant. web*, ročník 4, č. 2, Duben 2013: s. 119–158, ISSN 1570-0844. Dostupné z: <http://dl.acm.org/citation.cfm?id=2590215.2590217>
- [4] Marchiori, M.: The Mathematical Semantic Web. [online], [cit. 2019-04-22]. Dostupné z: [https://www.w3.org/People/Massimo/papers/2003/mathsemweb\\_mkm\\_03.pdf](https://www.w3.org/People/Massimo/papers/2003/mathsemweb_mkm_03.pdf)
- [5] Weisstein, E.; Ford, I.: The Semantic Representation of Pure Mathematics. [online], [cit. 2019-04-22]. Dostupné z: <https://blog.wolfram.com/2016/12/22/the-semantic-representation-of-pure-mathematics/>
- [6] Google: Introducing the Knowledge Graph: things, not strings. [online], [cit. 2019-04-22]. Dostupné z: <https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html>
- [7] Wolfram Research: Frequently Asked Questions. [online], [cit. 2019-04-22]. Dostupné z: <https://www.wolframalpha.com/faqs/>
- [8] 2010 Mathematics Subject Classification. [online], [cit. 2019-04-22]. Dostupné z: <https://mathscinet.ams.org/msc/msc2010.html>



## Definice pojmů

**Báze znalostí** technologie určená k ukládání komplexních informací pro použití počítačem

**Expertní systém** počítačový program poskytující expertní rady nebo rozhodnutí v konkrétní situaci

**Git** distribuovaný systém správy verzí

**Hypertextový odkaz** Reference na data (typicky webová stránka), která se zobrazí po kliknutí

**Jinja2** šablonovací systém pro programovací jazyk Python

**Ontologie** explicitní a formalizovaný popis určité problematiky, obsahuje definici pojmů a vztahů mezi pojmy

**Parser** syntaktický analyzátor

**Python** programovací jazyk

**Sémantická síť** báze znalostí reprezentující sémantické vztahy mezi pojmy

**Sémantická trojice** základní jednotka sémantické sítě, popisuje vztah mezi dvěma pojmy

**Sémantický web** technologie pro sémantickou anotaci informací na internetu

## A.1 Seznam použitých zkratk

**BI-ZMA** bakalářský předmět základy matematické analýzy

**BI-PKM** bakalářský předmět přípravný kurz matematiky

**CI/CD** Continuous Integration and Continuous Delivery

**ČVUT** České vysoké učení technické v Praze

**FIT ČVUT** Fakulta informačních technologií na ČVUT

**HTML** Hypertext Markup Language

**JSON** JavaScript Object Notation

**MathML** Mathematical Markup Language

**OWL** Web Ontology Language

**RDF** Resource Description Framework

**UTF-8** Unicode Transformation Format (8-bit variable-width)

**YAML** YAML Ain't Markup Language

---

## Obsah přiloženého CD

```
/
├── data
│   └── bi-zma.txt
├── semantic
│   ├── semantic.py
│   ├── __init__.py
│   ├── expressions
│   │   ├── parser.py
│   │   └── models.py
│   └── __init__.py
├── files
│   ├── parser.py
│   ├── models.py
│   ├── finder.py
│   └── __init__.py
├── elements
│   ├── parser.py
│   ├── models.py
│   └── __init__.py
├── templates
│   └── templates/template.html
├── text
│   ├── semanticke-site-matematickych-znalosti.tex
│   └── semanticke-site-matematickych-znalosti.pdf
├── output.html
├── README.md
├── requirements.txt
└── run.py
```