

**Univerzita Karlova v Praze**

**Filozofická fakulta**

**Ústav informačních studií a knihovnictví**

Studijní program: Informační studia a knihovnictví

Studijní obor: Informační studia a knihovnictví

## **Diplomová práce**

**Bc. Gabriela Ferbarová**

**Konceptuální struktury jako nástroj  
reprezentace znalostí**

**Conceptual Structures As a Tool  
for Knowledge Representation**

Praha 2016

Vedoucí práce: prof. RNDr. Jiří Ivánek, CSc.

Prohlášení:

*„Prohlašuji, že jsem diplomovou práci vypracovala samostatně, že jsem řádně citovala všechny použité prameny a literaturu a že práce nebyla využita v rámci jiného vysokoškolského studia či k získání jiného nebo stejného titulu.“*

V Praze, dne 31. července 2016

---

Jméno a příjmení

**Klíčová slova (česky):**

konceptuální struktury, konceptuální graf, reprezentace znalostí, ontologie, expertní systém, modelování uživatele,

**Klíčová slova (anglicky):**

Conceptual structures, Conceptual graph, Knowledge representation, Ontology, Konceptual modeling, Expert system, User modeling,

**Identifikační záznam:**

FERBAROVÁ, Gabriela. *Konceptuální struktury jako nástroj reprezentace znalostí = Conceptual Structures As a Tool for Knowledge Representation*. Praha, 2016-07-29, 148 s. Diplomová práce (Mgr.) Univerzita Karlova v Praze, Filozofická fakulta, Ústav informačních studií a knihovnictví. Vedoucí bakalářské práce Jiří Ivánek.

## **Abstrakt (česky)**

Konceptuální grafy jsou formálním jazykem reprezentace znalostí, které představil americký odborník na umělou inteligenci John F. Sowa na konci 70. let. Jsou syntézou formalistického a heuristického přístupu k umělé inteligenci a práci se znalostmi. Jejich prostřednictvím je možné reprezentovat význam a znalosti formou, která má preciznost logiky, je člověkem čitelná i srozumitelná a zároveň aplikovatelná v počítačové vědě všeobecně. Konceptuální grafy je možné vyjádřit prostřednictvím logiky prvního řádu, což z nich dělá kvalitní nástroj umělého odvozování. Jejich forma notace CGIF byla standardizována normou ISO/IEC 24707:2007 pod ISO jako jeden ze tří dialektů tzv. obecné logiky, která tvoří rámec skupiny jazyků založených na logice. Konceptuální grafy jsou navíc mapovatelné do jazyků reprezentace znalostí standardizovaných pro sémantický web; OWL a RDF(S). Tato práce ukotvuje teorii konceptuálních grafů do kontextu vědních oblastí lingvistiky, logiky i umělé inteligence. Představuje formalismus navržený Johnem F. Sowou i základní rozšíření, která se objevila v průběhu uplynulých dekad spolu s potřebou zefektivnit reprezentační vlastnosti grafů. Závěr práce přináší ilustrativní přehled implementačních využití konceptuálních grafů v praxi. [Autorský abstrakt].

## **Abstract (in English):**

Conceptual graphs are a formal knowledge representation language introduced by John F. Sowa, an American specialist on Artificial Intelligence, at the end of the seventies. They are the synthesis of heuristic and formalistic approach to Artificial Intelligence and knowledge procession. They provide meaning and knowledge in form, which is logically precise, human-readable and untestable, and it is applicable in the computing domain in general. Conceptual graphs can be expressed through a first-order logic, which makes them a quality tool for intelligent reasoning. Their notation CGIF was standardised by norm ISO/IEC 24707:2007 as one of the three dialects of Common logic, which frames the set of logic based on logic. Conceptual graphs are also mappable to knowledge representation languages standardised for the Semantic Web; OWL and RDF (S). This work introduces the conceptual graph theory in the context of scientific fields like linguistics, logic and artificial intelligence. It represents the formalism proposed by John F. Sowa and some extensions that have emerged over the past decades, along with the need for improvements to the representational properties of graphs. Finally, the work provides an illustrative overview of the implementation and use of conceptual graphs in practice. [Authors' abstract].

## Obsah

Předmluva .....	9
1 Úvod do problematiky teorie konceptuálních grafů .....	12
1.1 John Florian Sowa .....	12
1.2 Konceptuální struktury .....	13
1.3 Obecná charakteristika konceptuálních grafů.....	14
1.4 Konceptuální grafy jako most mezi heuristickým a formalistickým přístupem .....	16
1.5 Teorie ovlivňující vznik teorie konceptuálních grafů.....	17
1.5.1 Logická tradice Charlese Sandersse Pierce .....	18
1.5.2 Lingvistická tradice .....	23
1.5.3 Teorie sémantických sítí.....	25
1.5.4 Logická tradice: lingvistické vztahy .....	26
1.5.5 Logická tradice: unifikace .....	26
1.6 Vývoj konceptuálních grafů .....	27
2 Konceptuální grafy dle John F. Sowy.....	32
2.1 Definice konceptuálních grafů.....	32
2.1.1 Vlastnosti konceptuálních grafů.....	32
2.1.2 Formální definice.....	33
2.1.3 Speciální druhy konceptuálních grafů.....	34
2.2 Notace konceptuálních grafů (ISO/IEC 24707, 2007).....	34
2.2.1 Předpoklady validní konstrukce konceptuálního grafu: .....	35
2.2.2 Grafická forma.....	36
2.2.3 Forma lineárního textového kódu.....	37
2.3 Ontologie v konceptuálních grafech.....	37
2.3.1 Ontologie a její význam .....	37
2.3.2 Sowyův návrh ontologie nejvyšší úrovně.....	39
2.3.3 Koncept.....	45
2.3.4 Typ konceptu .....	46
2.3.5 Druhy konceptů.....	51
2.3.6 Koreferenční spojení .....	52

2.3.7	Konceptuální vztahy .....	55
2.3.8	Valence vztahu .....	55
2.3.9	Signatura vztahu.....	55
2.3.10	Typ vztahu .....	56
2.3.11	Příklady konceptuálních vztahů s rozdílnou valencí .....	58
2.3.12	Konvence směru hran v konceptuálních grafech.....	60
2.4	Lambda kalkulus a lambda výrazy v konceptuálních grafech.....	60
2.4.1	Definice lambda výrazu .....	60
2.4.2	Lambda výraz pro definici typů konceptu .....	63
2.4.3	Lambda výraz pro definici vztahů.....	64
2.5	Zahnížděné grafy a koncepty.....	64
2.5.1	Zahnížděné grafy .....	64
2.5.2	Zahnížděné a spolu – zahnížděné koncepty.....	65
2.5.3	Příklad zahnížděného konceptuálního grafu (Petersen, 2005):.....	65
2.5.4	Kontext .....	67
2.6	Základní logické operátory v konceptuálních grafech.....	68
2.6.1	Negace v konceptuálním grafu.....	68
2.6.2	Konjunkce v konceptuálním grafu.....	68
2.6.3	Disjunkce v konceptuálním grafu .....	69
2.6.4	Implikace konceptuálních grafů .....	69
3	Vybraná rozšíření teorie konceptuálních grafů.....	70
3.1	CGIF Conceptual Graph Interchange Format .....	70
3.2	Funkce.....	72
3.2.1	Definice:.....	73
3.2.2	Notace .....	73
3.2.3	Hierarchie typů závislostí .....	74
3.2.4	Aktory .....	75
3.3	Konjunktivní typy .....	75
3.4	Fuzzy logika a teorie fuzzy množin .....	76
3.4.1	Fuzzy typy konceptů a vztahů .....	77
3.5	Reprezentace času (Časové konceptuální grafy).....	80
3.6	Odvozování a konstrukce konceptuálních grafů .....	81

3.6.1	Kánonické grafy .....	82
3.6.2	Základní operace zobecnění a specializace .....	86
3.6.3	Odvozená pravidla formování grafů .....	90
3.6.4	Podpora (Support ) .....	95
4	Využití konceptuálních grafů v praxi .....	97
4.1	Expertní systémy .....	98
4.1.1	Budování expertního systému .....	98
4.1.2	Báze znalostí .....	99
4.1.3	Inferenční mechanismus .....	100
4.1.4	Příklad expertního systému v oboru zdravotnictví .....	101
4.1.5	Akvizice znalostí .....	101
4.2	Přehled využití konceptuálních grafů .....	105
4.2.1	Informační systémy .....	105
4.2.2	Inteligentní vyhledávání informací .....	105
4.2.3	Normy .....	106
4.2.4	Management znalostí .....	107
4.2.5	Scientometrie .....	107
4.2.6	Sémantický web .....	108
4.2.7	Modelování a modelování uživatele .....	109
4.3	Příklad z praxe: využití konceptuálních grafů pro funkci inteligentních agentů pro modelování uživatelů .....	111
4.3.1	Modelování uživatele .....	112
4.3.2	Inteligentní agenty .....	112
4.3.3	Teoretický základ a východiska .....	113
4.3.4	Inferenční mechanismus .....	116
4.3.5	Příklad odvozování v aplikaci Forum .....	121
4.4	Teoretický příklad systému NaZahrádce .....	127
4.4.1	Uživatelé .....	127
4.4.2	Expertní systém .....	128
4.4.3	Informační databáze .....	132
4.4.4	Další funkce .....	133
4.4.5	Zhodnocení .....	133

5	Závěr.....	134
	Seznam použité literatury .....	136
	Seznam obrázků .....	145
	Seznam vzorců .....	147



## Předmluva

Cílem této práce je představit konceptuální grafy jako nástroj reprezentace znalostí odborníkům z oboru informačního zaměření. Teorie konceptuálních grafů je ve světě známou a uznávanou disciplínou již po několik desetiletí. V českém prostředí je možné se o ní dozvědět více pouze z několika odstavců v učebnicích a skriptech umělé inteligence. Tyto texty jsou však většinou určené odborníkům z oblasti výpočetních technologií, od kterých se očekává vybavenost znalostmi z oboru aplikované matematické logiky, výpočetních algoritmů a zaměřují se spíše na aplikační užití konceptuálních grafů. Cílem této práce je představit konceptuální grafy v kontextu vědních oblastí informační vědy, filosofie, lingvistiky, logiky i umělé inteligence. Snahou bylo představit jejich formalismus takovým způsobem, který je jednoduše čitelný i pro odborníky nematematických oborů a na vybraných příkladech ilustrovat možnosti využití a aplikace konceptuálních grafů s ohledem na informační specializaci. Od čtenáře je očekávána základní orientace v oblasti informační vědy a reprezentace znalostí a pouze základní znalost operátorů predikátové logiky. Dle mého se jedná o velmi zajímavé pojetí reprezentace znalostí kloubící heuristický a formalistický přístup k práci se znalostmi a je možné jej využít v mnoha rozličných oblastech informační vědy.

Práce se liší od svého původního zadání, protože nezahrnuje tematiku formální konceptové analýzy, což je teorie, která se v dnešní době řadí mezi teorie konceptuálních struktur. Ukázalo se, že takové pojetí by bylo pro rozsah této práce příliš široké. Usoudila jsem, že bude vhodnější zaměřit se pouze na jednu z teorií a tu představit více do hloubky. Vzhledem k tomu, že konceptuálních grafů, na rozdíl od formální konceptové analýzy, je možno využít k odvozování znalostí, přišlo mi vhodnější vybrat si tento směr. Formální konceptová analýza je však krátce a zevrubně představena v první kapitole textu. Do práce také nebyla zahrnuta samostatná kapitola věnující se výhodám a nevýhodám konceptuálních grafů. Toto téma je obsaženo průběžně v dílčích kapitolách a tématu se krátce věnuji v závěrečné kapitole.

Práce je členěna do čtyř kapitol a páté závěrečné. První úvodní kapitola představuje konceptuální grafy v kontextu vědních oblastí, které měly vliv na jejich vznik a vývoj. Představuje také stručný vývoj teorie od první idey až po plnohodnotný formalizovaný jazyk

reprezentace znalostí. Obsahuje také stručné informace o autorovi teorie konceptuálních struktur Johnu F. Sowi. Dle mého je tato kapitola důležitá pro pochopení, z jakých tradic teorie konceptuálních grafů vychází a jak široké je jejich využití. Druhá kapitola shrnuje formalismus konceptuálních grafů, tak jak je představil John F. Sowa. Tato kapitola se věnuje sémantice i syntax konceptuálních grafů, představuje jejich možné notace a přibližuje tzv. lambda kalkulus a základní logické operace, což jsou základní prostředky k dovozování znalostí skrze konceptuální grafy. Tato kapitola také představuje teorii ontologií dle Johna F. Sowy, který jako jeden z prvních upozorňoval na význam ontologií pro informační a počítačovou vědu a umělou inteligenci. Ontologie společně s konceptuálními grafy mohou tvořit mocný nástroj reprezentace znalostí. Třetí kapitola představuje několik rozšíření konceptuálních grafů. Jedná se například o notaci konceptuálních grafů CGIF, jež byla standardizována normou ISO/IEC 24707:2007 pod ISO jako jeden ze tří dialektů tzv. obecné logiky, nebo o implementaci funkcí a fuzzy logiky pro znázornění funkčních závislostí a neurčitosti. Všechna tato dílčí rozšíření dle mého vedou k zefektivnění konceptuálních grafů jako jazyka reprezentace jazyků a posloužila k jeho lepší využitelnosti. Čtvrtá kapitola ilustruje možná použití konceptuálních grafů na výčtu aplikačních oblastí. Specifickým tématem je využití konceptuálních grafů v prostředí tzv. sémantického webu, který je momentálně považován za možnou budoucí podobu webu. Tato kapitola také obsahuje příklad z praxe, kdy bylo užito konceptuálních grafů pro modelování uživatele. Na tomto příkladu je také ilustrováno, jak funguje odvozovací mechanismus znalostí vystavěný na operacích s konceptuálními grafy. Tématiku modelování uživatele jsem si vybrala, protože koresponduje s tématem mé bakalářské práce. Kapitulu uzavírá popis teoretického systému NaZahrádce, kde se snažím čtenáři přiblížit veškerá možná využití konceptuálních grafů při návrhu informačního systému skrze jednoduchý příklad.

Při psaní práce jsem povětšinou vycházela z prací zahraničních autorů, a to především z monografií, jež k tématu napsal tvůrce konceptuálních grafů John F. Sowa. Dále jsem informace čerpala především z konferenčních sborníků a odborných článků zejména ze zahraničních periodik. Využívala jsem skripta a učebnice matematických oborů a umělé inteligence českých autorů pro vysvětlení některých pojmů a ujasnění správné terminologie. Dále jsem použila informace ze slovníků a encyklopedií a to jak českých, tak zahraničních.

Všechny použité zdroje jsou citovány podle aktuální verze normy ČSN ISO 690. Citované zdroje jsou uvedeny v části s názvem „Seznam použité literatury“ a jsou řazeny abecedně podle záhlaví. Odkazy na citované zdroje v textové části jsou označeny jmény autorů a rokem vydání publikace a jsou uzavřeny v hranatých závorkách. Způsob odkazování v textu na citovanou literaturu je proveden pomocí prvního popisného údaje a roku vydání – tzv. „Harvardským systémem“. Vzorce a obrázky v textu jsou číslovány. Ty, které byly převzaty z jiného zdroje pouze s menšími změnami, jako je překlad nebo adaptace obsahu, jsou řádně označeny odkazem na citované zdroje. Vzorce a grafy, jež jsou mým autorským dílem, nejsou označovány. Pro práci jsem zvolila bezpatkový styl písma. Vzhledem ke vzorcům a matematickým symbolům, pro které jsem zvolila matematický font Cambria Math, je tak text snadněji čitelný. Práce obsahuje 208738 znaků; tedy 116 normostran.

Chtěla bych velice poděkovat svému vedoucímu práce, panu profesorovi RNDr. Jiřímu Ivánkovi, CSc, za mnoho podnětných připomínek, rad a především trpělivost.

# 1 Úvod do problematiky teorie konceptuálních grafů

Reprezentace znalostí a odvozování je jednou z nejdůležitějších oblastí umělé inteligence. Cílem současného vývoje tohoto vědního oboru je nalézt efektivní techniky nezbytné pro vytvoření skutečně inteligentního systému. Základním cílem tohoto oboru je zakódovat lidské znalosti a způsob odvozování a uvažování v podobě symbolů tak, aby s nimi mohl pracovat počítačový systém, což by mělo vést k inteligentnímu chování stroje. Není cílem této práce hledat odpověď na otázku, co je inteligence a zda je skutečně možné, aby byl stroj inteligentní. Pro potřeby této práce je možné si inteligenci definovat jako schopnost jednat autonomně za účelem splnění nějakého úkolu nebo zodpovězení dotazu. v dnešní době velmi rychle narůstá množství dat a informací, se kterými je třeba manipulovat a zpracovávat je. I z tohoto důvodu se objevila myšlenka nové podoby webu; tzv. sémantického webu.

Problematiku skladování velkého počtu dokumentů, se zdá, podařilo více či méně uspokojivě vyřešit. Avšak o to je vyvíjen větší tlak na kvalitní reprezentaci znalostí a objevují se snahy vyvinout struktury a jazyky reprezentace znalostí optimalizované pro různé druhy odvozování, kompatibilní s moderním trendem využívání ontologií pro reprezentaci znalostí. Domnívám se, že by tyto požadavky mohl naplňovat formalismus konceptuálních grafů, který však vznikl před více než 30 lety. Tato kapitola seznamuje čtenáře s autorem konceptuálních grafů Johnem Florianem Sowou, představuje klíčové teorie ovlivňující podobu konceptuálních grafů a popisuje, jakým vývojem konceptuální grafy prošly, než byla ustálena jejich současná podoba.

## 1.1 John Florian Sowa

Autorem teorie konceptuálních struktur je americký vědec John Florian Sowa<sup>1</sup>, který se celý svůj život zabývá tématy umělé inteligence, expertními systémy a databázovými

---

<sup>1</sup> John F. Sowa se narodil roku 1940. John F. Sowa získal roku 1962 titul BS (Bakalář vědy) v matematice na Massachusettském technologickém institutu, magisterský titul z aplikované matematiky na Harvardu roku 1966 a doktorát z počítačové vědy na Univerzitě Vrije v Bruselu roku 1999 se svojí dizertační prací: Reprezentace znalostí: Logické, filozofické a počítačové základy. John F. Sowa strávil více než 30 let ve výzkumu a vývoji pro firmu IBM, kde se věnoval výzkumu a vývoji rozvíjejícího se oboru počítačové vědy: od překladačů, programovacích jazyků, přes architekturu systémů až po umělou inteligenci a reprezentaci znalostí. Po odchodu do důchodu v roce 2001 Sowa spolufinancoval VivoMind Research, LLC. s touto firmou pracuje na vývoji databázových a data-miningových technologií – konkrétně na vysokoúrovňové ontologii pro umělou inteligenci a automatizované porozumění přirozených jazyků. v průběhu let vedl kurzy a vyučoval v IBM Systems Research

dotazy propojenými s přirozenými jazyky. Mezi jeho zásadní práce patří rozvoj nových metod užití logiky a ontologie v systémech zaměřených na odvozování (reasoning) a jazykové porozumění. Jím navrhnutý jazyk konceptuálních grafů je jedním ze tří jazyků, které byly uznány v rámci ISO/IEC standardu 24707 (Common Logic). Za zajímavé považuji to, že Sowa ve svých pracích kombinuje znalosti a ideje nejen z velkého množství vědních oborů a disciplín, ale i z rozdílných epoch lidských dějin. Příkladem může být jeho implementace Aristotelových myšlenek a středověkých scholastiků do teorií Alfreda Northa Whiteheada (filosof, fyzik a matematik, z přelomu 19. a 20. století), do kterých navíc Sowa zahrnul i teorie databázového schématu a model analogického odvozování od islámského učenice Ibn Taymiyyah, žijícího na přelomu 12. a 13. století (Majudmar, Sowa, 2003).

## 1.2 Konceptuální struktury

Komunita, která se vytvořila okolo teorie konceptuálních struktur, tak jak jí představil John F. Sowa, se z počátku zaměřovala na teorii a aplikace konceptuálních grafů. Avšak brzy se k nim připojila komunita věnující se tzv. formální konceptuální analýze (FCA - Formal Concept Analysis). FCA původně vznikla jako podbor aplikované matematiky a je založená na matematizaci konceptů a hierarchií konceptů. v průběhu vývoje však formální konceptuální analýzu ovlivnily teorie filozofické logiky lidského myšlení a kognitivní strukturalismus. Zjednodušeně řečeno, formální konceptová analýza definuje způsob, kterak derivovat hierarchii konceptů, nebo můžeme říci formální ontologii z nějaké množiny objektů a jejich vlastností. Tyto hierarchie jsou většinou zobrazovány do tzv. konceptových svazů. Každý koncept hierarchie reprezentuje množinu objektů, které mají stejnou hodnotu určitých společných vlastností a koncept na nižší úrovni hierarchie obsahuje podmnožinu objektů konceptu nad ním. Hlavním cílem FAC je podporovat racionální komunikaci a zpracování znalostí. FAC se převážně využívá pro aplikace vytěžování dat, textů, strojové učení, znalostní management, atd.

Teorie konceptuálních grafů a formální konceptové analýzy sdílejí společný filozofický základ a mají mnoho společných aspektů. Jsou mezi sebou mapovatelné (Mineau, Stumme, Wille, 1999). Navíc se tyto přístupy mohou doplňovat. Kupříkladu tzv. kontextuální logika,

---

Institute, Binghamtonské univerzitě, Stanfordské univerzitě, Americké lingvistické společnosti a na Univerzitě Quebec v Montrealu. Je členem Americké asociace pro umělou inteligenci a Asociace pro pokrok v umělé inteligenci.

což je snaha o formalizaci tradiční filozofické logiky, která se opírá o definici konceptů, úsudků a závěrů (Wille, 2005). v teorii kontextuální logiky slouží formální konceptuální analýza k matematizaci konceptů, konceptuální grafy k matematizaci usuzování a kombinace grafů, FCA a deskripční logiky lze využít k matematizaci závěrů (Ganter, Stumme, Wille, 2005).

### 1.3 Obecná charakteristika konceptuálních grafů

Hned na úvod je vhodné rozebrat pojmy „konceptuální“ a „graf“. Slovo koncept je odvozeno z latinského *concupere*: pojmát, počít, uchopit a *conceptus*: pojetí, početí. Pojem koncept je synonymem slova „koncepce“, které je odvozeno z latinského *con-ceptio* (složeno z *con*: „dát dohromady“ a *capere*: brát, uchopit) (Holub, Lyr, 1992). Také je možné odvození od latinského *conceptio*, jež může znamenat: pojetí či pojem, způsob výkladu, chápání, základní hledisko, vůdčí idea, myšlenková osnova, hlavní záměr či myšlenka. Je možno tedy obecně shrnout, že termín z biologického hlediska znamená početí, oplodnění, ale jinak se užívá ve smyslu pojetí, rozvržení či představy. Pojem „graf“ je odvozeno od řeckého γράφω (*gráphō*), které znamená psát, zaznamenávat, načrtávat. Výrazu „graf“ se také užívá jako druhé části složených slov mající význam zapisující, zaznamenávající přístroj či zařízení. Výrazu graf se dále užívá ve významu schematického znázornění vztahů, závislostí, postupů, statistických údajů a v matematice se výrazu graf užívá pro orientovaný popisný objekt, jímž se vyjadřuje vztah, propojení mezi objekty (Foltýnek, Dannhoferová, 2011). Po syntéze těchto dvou pojmů se mi nejadekvátněji jevil překlad „popis myšlenkové osnovy“.

Konceptuální graf je grafickou reprezentací logiky založené na sémantických sítích umělé inteligence a existenciálních grafů Charlese Sanderse Pierce. Popudem ke vzniku konceptuálních grafů byl fakt, že se použití databázových systémů začalo rozšiřovat mezi běžné uživatele, kteří nebyli vybaveni dostatečnou odborností k přímé práci s databázovým systémem. Dle Sowy tito uživatelé potřebovali prostředek k pokládání databázových dotazů tak, aby nemuseli znát konkrétní způsob vázání databázových objektů (konkrétní vazby je nutné znát například při dotazování v jazyce SQL). Přišel tedy s pojmem konceptuálních grafů jako prostředku popisu dat a jejich vzájemných vazeb, který je ale abstrahován nad konkrétní způsob uložení dat v databázovém systému. Výhodou také je, že konceptuální grafy stojí někde mezi dotazováním v přirozeném jazyce a surovými databázovými dotazy, přičemž

z přirozeného jazyka je možné (byť stále poměrně obtížné) generovat konceptuální graf, a z konceptuálního grafu následně generovat databázový dotaz. (Sowa, 1976)

Konceptuální grafy byly tedy původně navrženy jako mezijazyk<sup>2</sup> uplatňovaný v relačních databázích. Přirozený jazyk a symbolická logika se dají považovat za univerzální reprezentace znalostí. Přirozené jazyky nastavují standard pro flexibilitu a expresivnost, protože se vypořádávají s každým aspektem lidského života včetně čehokoliv, co by kdy mohlo být vyřčeno v jakémkoliv umělém jazyce. Naopak symbolická logika nastavuje standard pro preciznost a všeobecnost formálních systémů, kdy jakýkoliv běžící počítačový program a dokonce počítač sám o sobě může být definován pomocí logiky. Zamýšleným účelem konceptuálních grafů je zajistit univerzální jazyk reprezentace znalostí, který kombinuje expresivní sílu přirozených jazyků s precizností symbolické logiky. (Sowa, 1992).

Jinak řečeno se konceptuální grafy postupem času vyvinuly v systém a staly se jazykem reprezentace znalostí, který se snaží vyjádřit význam formou, která má preciznost logiky, je člověkem čitelná i srozumitelná a zároveň aplikovatelná v počítačové vědě všeobecně. Díky tomuto základu jsou konceptuální grafy velmi vhodné pro zapracování do počítačových programů a v současné době se ve velké míře uplatňují v programech zaměřených na umělé usuzování. Jejich kvalita je dle mého názoru speciálně dána faktem, že nejsou závislé na jakékoliv striktně definované notaci (byť na jejich principech byly vytvořeny notace standardizované ISO normou). Vzniká tak možnost definovat vlastní notaci na základě předdefinovaného logického rámce a pravidel.

Konceptuální grafy mají tedy veškerou vyjadřovací sílu logiky, ale jsou více intuitivní a čitelné s bezproblémovým mapováním do přirozeného jazyka (Way, 1994). Díky jejich obecnosti a čitelnosti byly konceptuální grafy navrženy jako vnitřní normativní jazyk pro konceptuální schémata ANSI X3 výboru ISO pro slovníkový systém zdrojů informací (Committee on Information Resource Dictionary Systems). Navíc lze říci, že jsou ekvivalentem nebo naopak mocnějším nástrojem než většina jazyků reprezentace znalostí. Cokoliv zapsáno či vyjádřeno v rámcich nebo jazyce založeném na rámcich je možno vyjádřit pomocí konceptuálních grafů, avšak žádný z rámcových systémů není v reprezentaci tak flexibilní.

---

<sup>2</sup> Autorský překlad anglického termínu „interlanguage“

Například Schankovy (1969) konceptuální závislostní diagramy či teorie skriptů mohou být vyjádřeny skrze pomoci konceptuální grafy, které navíc dodávají Shankovým myšlenkám a přístupům preciznost a teoretický základ. Na mnoho populárních grafických notací a struktur od typových hierarchií po entitně vztahové modely a stavové diagramy může být nahlíženo jako na speciální typy konceptuálních grafů. Konceptuální grafy byly také vyvinuty jako sémantický ekvivalent grafické reprezentace k tzv. „znalostnímu výměnnému formátu“: Knowledge Interchange Format (KIF<sup>3</sup>). (Way, 1994)

#### 1.4 Konceptuální grafy jako most mezi heuristickým a formalistickým přístupem

Way (1994) tvrdí, že se Sowa ve své knize *Conceptual Structures* nesnažil pouze shrnout dosavadní znalosti o umělé inteligenci, ale především nalézt rovnováhu mezi dvěma přístupy k umělé inteligenci: tzv. neat and scruffy.

Rozdíl mezi neats a scruffies byl v oblasti umělé inteligence popsán Rogerem Schankem v polovině sedmdesátých let minulého století. Neat (můžeme říci čistý) přístup je založen na logice, sémantických sítích nebo grafech, čili technologiích, které se snaží úlohy řešit čistě a přesně (Kotowski, 2013). Scruffy (neboli špinavý) přístup předpokládá, že inteligence ji příliš komplikovaná, aby byla řešena homogenními systémy, které většinou využívají zastánci neat přístupu, a že v nich chybí empirické poznání. Neat přístup se více zaměřuje na logiku a ověřitelné a opakovatelné výstupy. Scruffy přístup naopak kritizuje neat přístup pro přílišný formalismus a nemožnost využití tohoto přístupu při snaze o napodobení zdravého rozumu žijících inteligentních tvorů. (Kolonder, 2002)

Sowa se ve své knize *Conceptual Structures* pokusil postavit most mezi přínosem logicismu a metod schématických modelů na straně jedné a expresivností heuristických metod na straně druhé. Konceptuální grafy jsou formálně definovány sémantikou teorie modelů, avšak zachovávají si flexibilitu a otevřenost scruffy metod.

---

<sup>3</sup> Konceptuální grafy jsou navrženy tak, aby měly přímé mapování z a do přirozeného jazyka. Jejich grafická notace je navržena tak, aby byla co nejlépe a nejjednodušeji čitelná člověkem. KIF byl však navržen tak, aby plnil jiné cíle. Byl navržen jako varianta predikátového kalkulu se znakovou řadou a syntax, jež zjednodušuje mapování z a do výpočetních systémů. Jedná se tedy o počítačový jazyk vyvinutý tak, aby umožnil systémům sdílet a znovu využít informace ze znalostních systémů. Je založen sémanticky na predikátové logice a syntakticky na LISP. To umožňuje využít KIF pro reprezentaci libovolné věty v predikátové logice prvního řádu. Tento jazyk byl definován v rámci projektu Ontolingua, který poskytuje možnost kooperativně tvořit ontologie, což umožňuje export ontologií do různých formalismů. (Sowa, 2008)



## 1.5 Teorie ovlivňující vznik teorie konceptuálních grafů

Teorie konceptuálních grafů se samozřejmě neobjevila jen tak z ničeho. Notace konceptuálních grafů spočívající v kruzích a obdélnících byla sice vymyšlena a zveřejněna až John F. Sowa, ale formální struktury jsou syntézou století výzkumu práce v oblasti logiky, lingvistiky, filozofie a umělé inteligence. Jejich autor John F. Sowa se inspiroval a stavěl na práci svých předchůdců i současných kolegů. Domnívám se, že pro hlubší pochopení konceptuálních grafů a především jejich přínosu pro studium Umělé inteligence bude užitečné se s těmito předchůdci a výsledky jejich výzkumu blíže seznámit.

Samotná publikace *Conceptual Structures* (1984), která nabízí první ucelené představení teorie konceptuálních grafů, obsahuje kapitoly představující poznatky z filozofie, matematiky, umělé inteligence, lingvistiky a psychologie, kterými se Sowa inspiroval, nebo které naopak zavrhl jako pro jeho práci nepoužitelné. Není cílem této práce představit kompletní přehled poznatků, které vedly Sowa k jeho teorii konceptuálních grafů. Avšak pro uchopení tématu je třeba uvést teorie a výsledky vědecké práce, které sám Sowa označuje pro svou teorii jako klíčové. Pro hlubší pochopení myšlenkových pochodů autora doporučuji přečtení úvodních kapitol publikace *Conceptual Structures*, byť jejich studium od svých čtenářů očekává alespoň základní znalosti výše zmíněných vědních oborů.

Je možno říci, že konceptuální grafy jsou jazykem reprezentace znalostí, jež vznikl syntézou několika rozdílných tradic. První z těchto tradic je teorie sémantických sítí, které se používaly pro strojový překlad a počítačnickou lingvistiku po více než 30 let. Druhým tradičním přístupem jsou techniky založené na logice jako je například unifikace, lambda kalkulus a Piercovy existenciální grafy. Třetí tradicí je lingvistický výzkum založený na závislostních grafech Luciena Tesnière a na různých formách pádové gramatiky a tematických rolí. Čtvrtou tradicí jsou diagramy datových toků a Petriho sítě, které poskytují počítačové mechanismy pro vtažení konceptuálních grafů do externích procedur databází. (Sowa, 1992).

Výsledkem je vysoce expresivní systém logiky, s přímým mapováním do a z přirozených jazyků. Lambda kalkulus podporuje definice pro taxonomické systémy a poskytuje základní mechanismy pro restrukturalizaci znalostních bází. Díky definičním mechanismům mohou být konceptuální grafy užity jako přechodné stádium mezi přirozeným jazykem a pravidly a rámci expertního systému; což je velmi důležitá vlastnost pro znalostní akvizici.

### 1.5.1 Logická tradice Charlese Sanderse Pierce

Logické základy teorie konceptuálních grafů se opírají především o práci filozofa a logika Charlese Sanderse Pierce. v roce 1883 Pierce vyvinul první lineární notaci pro logiku prvního řádu. s pozdějšími variacemi od Ernsta Schrödera, Giuseppe Peana a Bernarda Russela tak vznikl moderní systém predikátového kalkulu (Sowa, 1992).

- V Piercově původní notaci by věta „Farmář vlastní oslíka.“ vypadala takto:

$$\Sigma x \Sigma y (\text{farmář}x \bullet \text{oslíky} \bullet \text{vlastnit}xy)$$

Vzor. 1.1 Piercova notace věty „Farmář vlastní oslíka.“ (Sowa, 1992)

- Moderní notace predikátového kalkulu by stejnou větu zapsala následovně:

$$(\exists x) (\exists y) (\text{farmář}(x) \wedge \text{oslík}(y) \wedge \text{vlastnit}(x, y)).$$

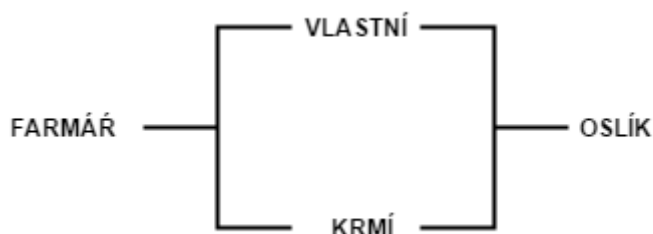
Vzor. 1.2 Notace věty „Farmář vlastní oslíka.“ v predikátovém kalkulu (Sowa, 1992)

Je vidět, že obě formule jsou ekvivalentní a došlo pouze ke změně v závorkách a v zápisu symbolů operátorů, kdy  $\Sigma$  je nahrazeno  $\exists$  a symbol  $\bullet$  modernějším  $\wedge$ . i přesto Pierce cítil, že predikátová notace pro logiku je příliš složitá a v roce 1982 vytvořil tzv. relační grafy, ve kterých čáry, jež Pierce nazývá linky identity, zastupují existenciální kvantifikátor a věta „Farmář vlastní oslíka.“ vypadala takto:

$$\text{FARMÁŘ} \text{ --- } \text{VLASTNÍ} \text{ --- } \text{OSLÍK}$$

Vzor. 1.3 Relační graf znázorňující větu „Farmář vlastní oslíka.“ (Sowa, 1992)

- Komplexnější relační graf, znázorňující komplikovanější větu: „Farmář vlastní a krmí oslíka.“ by vypadal následně:



Obr. 1.1 Relační graf znázorňující větu „Farmář vlastní a krmí oslíka.“ (Sowa, 1992)

- V predikátovém kalkulu by tato věta vypadala následně:

$$(\exists x) (\exists y) (\text{farmář}(x) \wedge \text{oslík}(y) \wedge \text{vlastnit}(x, y) \wedge \text{krmit}(x, y)).$$

*Vzor. 1.4 Notace věty „Farmář vlastní a krmí oslíka.“ v predikátovém kalkulu (Sowa, 1992)*

Je zřejmé, že relační grafy mohou vyjadřovat konjunkce a existenciální kvantifikátory. Pierce se pokoušel nalézt vhodné označení uzlů a hran speciálními symboly, které by mohly označovat negaci, disjunkci a univerzální kvantifikátor. Po mnoho let však nebyl schopen najít obecný způsob, jak reprezentovat všechny možné existující kombinace těchto výše zmíněných operátorů. v roce 1987 (publikováno až v listopadu 1906 v časopise *The Modists*) Pierce však vymyslel pro svou novou verzi existenciálních grafů jednoduchou, avšak geniální inovaci: zobrazovaný prostor uzavřel do oválu. Standartní operátor pro tento ovál (pokud neobsahoval jiné operátory) byla negace.

Tyto nové grafy tak oplývaly jednoduchým, avšak velmi silným a funkčním mechanismem tzv. zahrnutých grafů, jež byly součástí větších grafů. Takto vzniklé zahrnuté grafy nebo je možno říci kontexty, mu umožnily rozšířit kapacitu grafů na kompletně celou logiku prvního řádu a následně také i na modální logiku a logiku vyššího řádu (Sowa, 2008). Právě mechanismus zahrnutých grafů následně Sowa využil ve svých konceptuálních grafech. Sowa však Piercovy ovály nahradil obdélníky. Dle jeho vlastních slov „Obdélníky se dají lépe vnořit (zahrnut) nežli ovály a co je důležitější: každé kontextové okno může být interpretováno jako konceptuální okno, jež obsahuje zahrnutý konceptuální graf.“ (Sowa, 2008)



V grafu na Obr. 1.2 značí čára existenciální kvantifikátor a každý ovál znázorňuje negaci. Zahníždění dvou oválů tedy reprezentuje implikaci jelikož  $\sim (p \wedge \sim q)$  je ekvivalentem  $(p \supset q)$ . Větší vnější ovál značící část implikace „Jestliže (if)“ obsahuje jednu čáru pro farmáře a jednu pro oslíka s tím, že jsou obě tyto čáry prodlouženy až do menšího oválu značící „Pak (then)“. Tyto dvě čáry korespondují s proměnnými  $x$  a  $y$  a v predikátovém kalkulu by věta vypadala takto:

$$\sim (\exists x) (\exists y) (\text{farmář}(x) \wedge \text{oslík}(y) \wedge \text{vlastnit}(x, y) \wedge \sim \text{krmit}(x, y)).$$

*Vzor. 1.5 Notace věty „Pokud má farmář oslíka, tak ho krmí.“ v predikátovém kalkulu (Sowa, 1992)*

Když je užito operátoru  $\supset$  pro implikaci, musí být existenciální kvantifikátor pro farmáře a oslíka nahrazen univerzálním kvantifikátorem na začátku formule:

$$(\forall x) (\forall y) ((\text{farmář}(x) \wedge \text{oslík}(y) \wedge \text{vlastnit}(x, y)) \supset \text{krmit}(x, y)).$$

*Vzor. 1.6 Notace věty „Pokud má farmář oslíka, tak ho krmí.“ v predikátovém kalkulu 2 (Sowa, 1992)*

Pokud se však tato formule přeloží do přirozeného jazyka, bude znít nepřirozeně: „Pro všechna  $x$  a pro všechna  $y$ , pokud  $x$  je farmář a  $y$  je oslík a  $x$  vlastní  $y$ , tak  $x$  krmí  $y$ .“ Pro reprezentaci implikací mají Piercovy ovály lepší schopnost mapovat do angličtiny (a v tomto případě i do českého jazyka) nežli predikátový kalkulus s operátorem  $\supset$ .

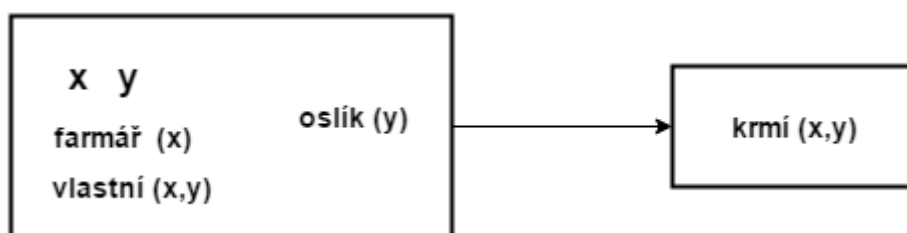
### Struktura reprezentace diskurzu

Během vývoje svých existenciálních grafů se Pierce snažil nalézt co nejjednodušší primitiva<sup>4</sup>, která by zachovala veškerou logiku, avšak nesnažil se zachovat lingvistickou přirozenost jazyka. Zajímavé je, že Piercovy zahnížděné kontexty jsou izomorfní<sup>5</sup> se strukturou reprezentace diskurzu (DRS) Hanse Kampa (Sowa, 2008). Hans Kamp vyvinul DRS

<sup>4</sup> Primitiva jsou dále neanalyzovatelné sémantické jednotky, z nichž lze poskládat kterýkoli jiný jazykový význam. (Kadlecová, 2014)

<sup>5</sup> Izomorfismus je velmi obecný koncept, který se objevuje v několika oblastech studia matematiky. Slovo je odvozeno z řeckého *iso*, jež znamená „rovný“ a *morphosis*, což znamená formovat nebo tvarovat. **Izomorfismus** je zobrazení mezi dvěma matematickými strukturami, které je vzájemně jednoznačné (bijektivní) a zachovává všechny vlastnosti touto strukturou definované. Neformálně řečeno se jedná o mapování, kdy každému prvku první struktury odpovídá právě jeden prvek struktury druhé a jsou zachovány i vztahy k ostatním prvkům. (Weisstein, 2016)

ve snaze reprezentovat kvantifikaci a anaforu<sup>6</sup> v přirozených jazycích (Sowa,1992). DRS je mentální reprezentací vytvářenou posluchačem na základě odvíjející se rozmluvy (diskurzu). DRS sestává ze dvou částí: z množiny individuí (diskurzních značek), které reprezentují diskutované objekty a z množiny vztahů mezi těmito individui (podmínek-conditions), které kódují informace, jež se akumulovaly v těchto diskurzních značkách. DRS tedy obsahuje informace o individuích, která se mohou například stát předměty anaforických odkazů.



Obr. 1.4 DRS pro větu „Pokud má farmář oslíka, tak ho krmí.“ (Sowa, 1992)

Na Obr. 1.4 proměnné **x** a **y** reprezentují diskurzní značky, které korespondují s čárami v Piercových grafech. Dva rámečky jsou kontexty a šipka znázorňuje implikaci. Implikace v Kampově podání nekoresponduje se symbolem  $\supset$  v predikátovém kalkulu, protože Kamp pracuje s doplňujícím předpokladem, že proměnné jsou v konsekventu<sup>7</sup> v rozsahu diskurzních značek v antecedentu<sup>8</sup>. (Sowa, 1992)

<sup>6</sup> Anafora je lingvistická entita, která indikuje, že existuje nějaká referenční vazba k jiné lingvistické entitě ve stejném textu. (Tognini-Bonelli, E. 2001.) Příklad: „Pokud má farmář **oslíka**, tak **ho** krmí.“ Slova „**oslík**“ a „**ho**“ označují stejnou entitu. Slovo „**ho**“ je anaforický odkaz někdy také nazývaný anafor.

<sup>7</sup> Druhý člen implikace se nazývá konsekvent. Konsekventu se říká nutná podmínka, čímž je reflektováno to, že implikace je pravdivá, pokud je její konsekvent pravdivý (Kuchyňka a Ráclavský, 2014, str. 9). Konsekvent neboli důsledek je druhou částí složeného výroku nazývaného implikace následující po „pak“ (Štěpán, 2011, str. 28).

<sup>8</sup> První člen implikace se nazývá antecedent, kterému se říká dostatečná podmínka, čímž je reflektováno to, že souvětí tvaru implikace může být pravdivé, ačkoli není naplněna podmínka z antecedentu (Kuchyňka a Ráclavský, 2014, str. 9). Antecedent neboli příčina je první částí složeného výroku nazývaného implikace a následuje po „jestliže“ nebo „pokud“ (Štěpán, 2011, str. 28).

Kampova primitiva jsou stejná jaká používal Pierce: výchozí je existenciální kvantifikátor a výchozí booleovský operátor je konjunkce. Negace je vyjádřena kontextovým rámečkem a implikace je vyjádřena dvěma kontexty. Jak je vidět na Obr. 1.2 Piercovy zahnížděné kontexty umožňují kvantifikátorům v antecedentu implikace zahrnout konsekvent do jejich působnosti. Kamp, který v Obr. 1.4 spojuje své rámečky pomocí šipek, formuluje přesně stejný předpoklad o působnosti kvantifikátorů. (Sowa, 2008)

Kamp analyzoval diskurz mnohem více do hloubky a formuloval pravidla pro interpretaci anaforických odkazů, avšak jakékoliv pravidlo vyjádřeno pomocí DRS notace lze aplikovat do notace existenciálních i konceptuálních grafů (Sowa, 2008). Pierce vyvinul své grafy 80 let před Kampem a pro úplně jiný účel. Pierce se snažil vyvinout formalismus pro zachycení logiky a Kamp se snažil reprezentovat anafory a roli kontextu v přirozených jazycích. Avšak Sowa pokládá za důležité fakt, že byť Kamp i Pierce pracovali nezávisle a na odlišných problémech, ke kterým přistupovali každý ze zcela jiného úhlu pohledu, výsledkem jsou izomorfní struktury. Sowa z toho vyvozuje, že takovéto struktury musejí být významné. (Sowa, 1992, 2008; Way, 1994)

### 1.5.2 Lingvistická tradice

Pierce i Kamp byli pomocí svých zahnížděných grafů schopni reprezentovat celkovou strukturu logiky a diskurzu, avšak nezačleňovali do kontextu jemnější lingvistické detaily. Aby dosáhl takové úrovně detailnosti, vyvinul Lucien Tesnière (1959) grafové struktury pro svůj systém závislostní gramatiky<sup>9</sup>. Výzkum Luciena Tesniéra měl významný vliv na lingvistické teorie, které kladou větší důraz na sémantiku nežli na syntax. Igor Aleksandrovič Mel'čuk využíval Tesniérovu grafové struktury více než 30 let pro své studium lexikálních vzorů ruského a francouzského jazyka. Roku 1964 převzal Tesniérovu grafové struktury pro svůj výzkum v oblasti strojových překladů a počítačnické lingvistiky David Glenn Hays a následně představil tzv. teorii závislostní gramatiky jako formální alternativu pro Chomského syntaktické notace (Sowa, 2000, 2015). Klein a Simmons (1963) implementovali teorii závislostní gramatiky do jejich systému strojového překladu. Výzkum Davida Hays

---

<sup>9</sup> **Závislostní gramatika**, nazývaná též jako závislostní syntax, představuje lingvistický směr popisu struktury věty založený na vztahu závislosti mezi slovními tvary, resp. větnými členy ve větné struktuře. Předmětem studia závislostní gramatiky jsou závislostní vztahy mezi členy řídicími a členy závislými, jež jsou ve větě reprezentovány jednotlivými slovními tvary (Motalová, 2014). „Za hlavní kritérium pro rozlišení členu řídicího a závislého (tedy pro určení orientace závislostního vztahu) je možné považovat v prototypickém případě syntaktickou vypustitelnost závislého členu v endocentrických konstrukcích“ (Hajičová, 2006, str.4).

a Simmonse do značné míry ovlivnil Rogera Schanka, který převzal závislostní přístup a přesunul důraz ze syntaktických závislostí na závislosti konceptuální – nebo můžeme říci kladl větší důraz na koncepty nežli na slova. (Sowa, 1992, 2015)

Sowa poukazuje na to, že i přes počítačovou i lingvistickou sofistikovanost jsou grafy Tesniéra, Melčuka, Hayse i Schanka na stejné logické úrovni a mají stejnou vyjadřovací sílu jako původní Piercovy závislostní grafy z roku 1882. Sowa tvrdí, že jediné logické operátory, které mohou reprezentovat, je konjunkce a existenční kvantifikátor. Sowa dodává, že i když mají některé z těchto grafů uzly označené jinými logickými operátory jako je například disjunkce, negace nebo univerzální kvantifikátor, nejsou schopny správně vyjádřit jejich rozsah. (Sowa, 2015)

Například Tesniérovi Sowa vytýká, že se zaměřoval na syntaktické vztahy mezi slovy a frázemi, ale neodlišoval různé druhy sémantických vztahů. Příkladem jsou 4 věty, ve kterých by Tesniérové závislostní grafy měly neoznačené hrany spojující sloveso s podmětem a předmětem:

- Tomáš postavil dům.
- On vlastní dům.
- On má rád dům.
- On prodal dům.

Syntakticky mají tyto věty stejný vzorec: podmět-sloveso-předmět. Sémanticky se však jejich základní vztahy velmi liší. „Stavění“ a „prodávání“ jsou akce, jejichž podměty jsou činiteli (agenty), avšak „vlastnit“ a „mít rád“ jsou stavy, které žádné činitele (agenty) nemají. „Mít rád“ je mentální stav, jehož podmětem je prožívající osoba. „Vlastnit“ je legální stav, jehož podmět je pouze ve stavu vlastnictví. Předmět akce „stavění“ neexistuje, dokud není akce vykonána; je tedy výsledkem akce. Avšak pro ostatní slovesa je předmět pasivním účastníkem, který existuje před akcí nebo stavem a v anglickém jazyce se nazývá *patient*. Sowa tvrdí, že pro odlišení těchto sémantických vztahů je nezbytné, aby hrany v grafu byly označeny jako agent či patient nebo zkratkami AGNT a PTNT, což následně vtělil do své teorie konceptuálních grafů. (Sowa, 1992)



### 1.5.3 Teorie sémantických sítí

Jak bylo řečeno v úvodu, pro vývoj teorie konceptuálních grafů byl důležitý výzkum sémantických sítí. Sémantická síť je grafová struktura sloužící reprezentaci znalostí. Jedná se o orientovaný graf tvořený uzly a hranami. Uzly reprezentují jednotlivé objekty popisovaného světa a hrany spojují tyto uzly pak reprezentují vztahy mezi těmito objekty. Počítačová implementace sémantických sítí byla vyvinuta v rámci rozvoje umělé inteligence a strojového překladu, ale dřívější verze sémantických sítí byly užívány ve filozofii, psychologii a lingvistice. Sémantické sítě začaly být rozvíjeny na konci 60. let a byly prvně implementovány ve snaze definovat typy konceptů a vzorce vztahů pro strojové překlady. (Sowa, 2015) První publikované užití pojmu „sémantická síť“ se objevilo v návrhu Richarda Richense na předprogramování systému pro strojový překlad:

*„Nyní mám na mysli konstrukci interlingua, ze které jsou odstraněny všechny strukturální zvláštnosti základního jazyka a zbyde to, co bych nazval „sémantickou sítí obnažených idejí.“ (Richens, 1956)*

Margaret Masterman a její kolegové z Cambridge Language Research Unit (CLRU) navrhli první systém, který se dá nazvat sémantickou sítí a který v roce 1961 během Mezinárodní konference strojových překladů jazyků a aplikovaných jazykových analýz (International Conference on Machine Translation of Languages and Applied Language Analysis) Margaret Masterman (1961) představila veřejnosti. Jednalo se o notaci založenou na grafech, jež se nazývala „sémantické sítě“, které v sobě již zahrnovaly mřížky konceptuálních typů (Sowa, 1992, 2008, 2015).

V témže roce představil Silvio Ceccato (1961) teorii korelativních sítí, jež obsahovaly na 56 různých typů vztahů (relací) včetně podtypu, instance, část celku, příbuzenských vztahů a různých typů atributů. Krom lingvistických vztahů zahrnul do svého seznamu i vyjádření vztahů místa, času a struktur jako je výše zmíněný typ vztahu „část celku“ nebo „prvek z kolekce“. Ceccato použil korelace a vzory při vedení syntaktického analyzátoru (parser) a při řešení vzniklých nejasností. Ceccato byl také první, kdo ve svém systému grafů označil vztahy typem. (Sowa, 2015). Sowa (1992) však Ceccata kritizuje za neodlišování a smíchání lingvistických vztahů nižší úrovně se vztahy vysoké úrovně, jež jsou mnohem

komplexnější. Lingvisté Fillmore a Gruber následně tyto vztahy analyzovali do větší hloubky a klasifikovali je systematictěji. Jejich teorie sémantických pádů a tematických relací se staly běžnou součástí lingvistické teorie a měly velký vliv na vyvíjející se teorii závislostní gramatiky.

David Hays (1964) prezentoval dependenční či je možno říci závislostní grafy, jež formalizovaly notace vyvinuté lingvistou Lucienem Tesnièreem (1959). Rané grafové notace reprezentovaly racionální struktury přirozených jazykových sémantik, ale žádná z nich nemohla plně vyjádřit logiku prvního řádu.

#### 1.5.4 Logická tradice: lingvistické vztahy

V teorii konceptuálních grafů jsou lingvistické vztahy vnímány jako primitiva, ale s možností komplexnější a složitější vztahy následně definovat. Tento fakt je důležitý především při implementaci konceptuálních grafů do expertních systémů, protože se v nich povětšinou pracuje se složitějšími typy vztahů, kdy mají pravidla a rámce tendenci užívat komplexních vztahů jako je např.: původní-cena-v-dolarech. Takto komplikované vztahy jsou na hony vzdáleny od lingvistických primitiv. Avšak i tyto komplexní koncepty a vztahy vyššího řádu mohou být vyjádřeny a definovány pomocí jednodušších vztahů řádu nižšího. Konceptuální grafy pro takovéto vyjádření používají funkce nebo lambda kalkulus, které budou vysvětleny dále.

#### 1.5.5 Logická tradice: unifikace

Metoda porovnávání vzorů či šablon (pattern matching) byla vždy důležitou technologií umělé inteligence. Sowa (1992) tvrdí, že grafy jsou velmi dobrým nástrojem na podporu nalezení shodných vzorů v komplexních strukturách. v roce 1965 představil John Alan Robinson svůj unifikační algoritmus, který původně sloužil ke strojovému dokazování tvrzení. Robinsonův objev umožnil prudký rozvoj automatického dokazování teorémů a logického programování (Duží, 2012). Následně byla tato metoda adaptována na unifikaci<sup>10</sup> grafů. Krom schopnosti reprezentovat logiku a jazyk vyžaduje počítačový systém spojení s dalšími počítačovými zařízeními a s okolním světem. Toho je docíleno pomocí speciálních

---

<sup>10</sup> Metoda, kdy jsou porovnány a propojeny odpovídající koncepty a vztahy z dvou rozdílných ontologií. Tato metoda umožňuje mapovat odvozovací a výpočetní operace vyjádřené v jedné ontologii do odpovídajících operací ve druhé ontologii (Sowa, 2000, str. 497).

uzlů aktorů (viz dále) a algoritmu, který je inspirován teorií Petriho sítěmi, což je matematický nástroj pro modelování a simulaci diskrétních systémů.(Sowa, 1992)

## 1.6 Vývoj konceptuálních grafů

Je třeba si uvědomit, že uběhlo téměř 40 let od chvíle, kdy se John F. Sowa začal zabývat svou teorií konceptuálních grafů a kdy ji poprvé uveřejnil. Vzhledem k tomu, že se jedná o stále živé téma, je logické, že teorie konceptuálních grafů prošla vývojem. Pohled na definici formalismu i využití konceptuálních grafů se v průběhu doby vyvíjel a měnil. v průběhu několika dekád bylo navrženo a implementováno několik různých verzí konceptuálních grafů(Sowa, 2008). Jak bylo popsáno výše, sám autor svou teorii několikrát přepracoval, doplnil či upravil. Také se objevilo mnoho odborníků, kteří doplnili či obohatili původní Sovaův koncept. Tento výzkum v oblasti konceptuálních grafů přinesl nové techniky usuzování, reprezentace znalostí a sémantiky přirozených jazyků.

Jak popisuje předchozí kapitola, konceptuální grafy vznikly především syntézou logiky a lingvistiky: logiky založené na grafech Charlese Sanderse Pierce, unifikaci a lambda kalkulu a lingvistice založené na závislostní gramatice, závislostních grafech a pádové gramatice. Tato syntéza však samozřejmě nevznikla najednou. Poprvé byla teorie konceptuálních grafů uveřejněna v roce 1968 jako seminární práce pro Marvinu Minskeho v rámci přednášek o umělé inteligenci na Massachusettském institutu technologie (Sowa, 1992). Tato první verze představila notaci skládající se z obdélníků, kruhů a využívala zahnížděné obdélníky pro vyjádření vedlejších vět.

Sám autor (1992) tvrdí, že jediné grafové systémy, které tehdy ovlivnily jeho první verzi teorie byly Haysovy závislostní grafy (Hays, 1964) a sémantické sítě M. Rosse Quilliana (1968). Sowa však nepovažoval systémy, které v té době studoval, za dostatečně flexibilní, aby naplňovaly jeho představu notace reprezentace znalostí. Jednoho dne ho však napadlo, že takováto notace by mohla mít formu podobnou stavebnici Tinkertoys, což je americká dětská stavebnice sestávající se špulek, kol a dalších dílů, které se dají volně skládat do různých konstrukcí. Sowa chtěl jazyk, který by obsahoval rozdílně formované propojené bloky, které by bylo rekombinovat různým způsobem tak, aby vytvořily nové struktury. Ve svém prvním návrhu využil myšlenky diagramových šablon ke konstrukci notace na bázi stavebnice skládající se z kruhů a obdélníků, která znázorňovala vedlejší věty v anglickém

jazyce vnořenými obdélníky. Tato první verze však rozhodně neobsahovala pravidla tvorby ani Pierccovy existenciální grafy. (Way, 1994)

Sowa se začal konceptuálními grafy jako jazykem reprezentace znalostí pro design databází vážněji zabývat až v 70. letech během své práce pro výzkumný institut IBM (Way, 1994). Do své první tištěné verze teorie konceptuálních grafů (Sowa, 1976), která se zabývala využitím konceptuálních grafů pro databázové rozhraní Sowa implementoval pravidla tvorby a grafovou unifikaci, které vycházely z poznatků přednášek Alana Robinsona na půdě IBM. Sowa dále připojil uzly aktorů (tzv. actor nodes), což jsou uzly speciálních typů vztahů představujících funkce. Tato implementace přispěla k vylepšení funkčnosti databázových dotazů. Následně Sowa zavedl terminologii, jež byla v souladu s teorií relačních databází, kdy na operace *spojení (join)* a *projekci (projection)* v konceptuálních grafech bylo nahlíženo jako na intenzionální protějšky operací *spojení a projekce* v relačních databázích.

Tato verze však ještě neobsahovala simplifikaci jako jedno z normativních pravidel tvorby. Místo toho obsahovala tzv. pravidlo odloučení, které dovolovalo odstranit konceptuální vztahy z grafu (Way, 1994).

Trvalo dalších 8 let, než Sowa uveřejnil svoji teorii v knize *Conceptual Structures* (1984), která představuje konceptuální grafy jako syntézu formalismu užívaného v umělé inteligenci a kognitivních věd. Během těchto let Sowa objevil a začlenil do své teorie Pierccovy existenciální grafy a pravidla tvorby získala dnešní podobu. Postupně přibyl lambda kalkulus a „herně-teoretická“ sémantika Jaakko Hintikka (Kuchyňka a Ráclavský, 2014). Vzhledem k tomu, že se Roger Schank vždy formalismu vyhýbal, nepřispěla jeho práce do teorie konceptuálních grafů žádným formálním nástrojem. Avšak jeho výzkum se výrazně podepsal na způsobu, jakým konceptuální grafy podporují aplikace umělé inteligence. (Way, 1994)

První dvě kapitoly knihy nastiňují filozofické a psychologické problémy a otázky spojené s umělou inteligencí a zbytek knihy rozvíjí formální teorii konceptuálních grafů, kterou aplikuje na logiku, lingvistiku a znalostní inženýrství. Poslední kapitola analyzuje limity konceptuálního myšlení a věnuje se otázce hledání kompromisu mezi diskrétními symboly a kontinuálním užíváním metafor a mezi logickým a asociativními způsoby myšlení (Way, 1994). Kniha odstartovala celosvětové hnutí, které začalo nenápadně několika workshopy

a které se postupně proměnily v konference s mezinárodní účastí. Konference International Conference on Conceptual Structures je zaměřena obecněji na konceptuální struktury a v srpnu 2016 proběhne ve Francii její dvacátý druhý ročník.

Když byla kniha *Conceptual Structures* v roce 1983 dokončena, neobsahovala některé nové a zásadní objevy z oblasti lingvistiky, umělé inteligence a teorie diskurzu, které jsou pro konceptuální grafy relevantní. Mezi tyto objevy patří, kromě zobecněných kvantifikátorů, již dříve v textu popsaná struktura reprezentace diskurzu Hanse Kampa (DRS), kdy Kamp vyvinul způsob systematického zacházení s kontexty, referenty a rozlišování anafor v propojeném diskurzu. Barwise a Perry (1983) zveřejnili svou teorii situační sémantiky, jež se stala jednou z nejdůležitějších teorií řešící významy v přirozených jazycích. Formální notace definovaná Barwisem a Perrym pro jejich situační sémantiku byla však dle Waye (1994) velmi nepraktická a pro velké systémy špatně čitelná. Pomocí konceptuálních grafů je však možno zformovat velmi rozvinutou notaci schémat situační sémantiky, která je schopna objasnit sémantické vztahy bez přílišné komplikovanosti a mnohomluvnosti.

Později se objevily objektově orientované programovací jazyky reprezentující paradigma pro výpočty a simulace komplexního chování interagujících objektů. Objektově orientované programování sebou nepřináší jen jednoduché doplnění několika nových nástrojů přidaných do programovacích jazyků. Jedná se spíše o přístup k designu procesu dekompozice komplexních problémů a vyvíjení programových řešení. Většina nástrojů, kterou konceptuální grafy potřebují pro modelování objektově orientovaných systémů, je dostupná již od začátku. Mechanismy dědičnosti (inheritance), které jsou pro objektově orientované programování nezbytné, jsou vlastně základním nástrojem konceptuálních grafů. Zapouzdření popisu objektů je možno dosáhnout za pomoci užití kontextů. Jiné nástroje, které jsou nezbytné pro modelování jiných aspektů objektově orientovaných jazyků, jako jsou například objektové třídy, objektové instance nebo zprávy, vyžadují explicitní konvence<sup>11</sup> pro použití konceptuálních grafů, ale opět není třeba žádného zvláštního rozšíření teorie či notace, aby se dalo užít konceptuálních grafů. (Way, 1994; Sowa, 2000)

---

<sup>11</sup> Sowa tyto konvence popisuje ve svém článku *Logical foundations for representing object-oriented systems* (Sowa, 1993).

Všechny tyto pokroky se objevily až po zveřejnění teorie konceptuálních grafů, ale i přesto síla Piercovy logiky a Sowova formalismu dokázala, že všechny tyto nové poznatky mohou být včleněny do konceptuálních grafů bez jakéhokoliv rozšíření původní teorie. Jejich implementace nevyžadovala žádný zásadní zásah do formalismu konceptuálních grafů. Kontexty konceptuálních grafů byly založeny na Piercových existenciálních grafech, jenž jsou s těmi Kampovými izomorfní, lambda kalkulus poskytl základ pro reprezentaci zobecněných kvantifikátorů a situační sémantika vedla pouze k objasnění významu grafů bez nutnosti zásahu do vnitřní struktury a formalismu (Sowa, 1992; Way, 1994).

Fakt, že v základní teoretické struktuře konceptuálních grafů k žádným velkým změnám nedošlo, je velmi zajímavý. Jednalo se spíše o progresivní prohlubování a hledání nových možností aplikace a interpretace originální teorie. To, že se v průběhu uplynulých let formalismus měnil tak málo, není způsobeno ani tak tím, že by teorie konceptuálních grafů byla zkosnatělá či zastaralá. Tento fakt naopak značí to, že Sowa ve své knize (1984) položil solidní a komplexní základ, na kterém je možné dále stavět a jeho teorii konceptuálních grafů nejrůznějším způsobem implementovat.

Sám autor k tomu říká: „*Mnou navržená notace není součástí formální teorie. Žádná definice, předpoklad, teorém nebo důkaz na ni v mém díle nikdy neodkazovaly. Je možno užít jakékoliv notace, která koresponduje s osobními stylistickými preferencemi. Podstatné je, že teorie konceptuálních grafů nebyla navržena tak, aby vyloučila či nahradila ostatní teorie. Naopak, aby je spíše adaptovala a demonstrovala, jakým způsobem jsou ve vzájemném vztahu.*“ (Sowa, 1992).

Dnes je již sémantika základních i rozšířených konceptuálních grafů definována formálním mapováním v rámci ISO standardem 247007 pro Common Logic. (Sowa, 2001, 2008) Avšak to, že jsou konceptuální grafy formalismem reprezentace znalostí založeným na logice znamená, že je podstatná spíše jejich forma a mechanismus než obsah. Tento fakt činí z konceptuálních grafů relativně univerzální nástroj a dává velký prostor k experimentům s nejrůznějšími druhy domén, ontologií, datových struktur a inferenčních strojů (Way, 1994).

Jak bylo zmíněno, konceptuálními grafy se zabývá relativně široká vědecká veřejnost a vzniklo tak mnoho formálních i neformálních rozšíření, jež byly aplikovány v nejrůznějších oblastech: expertní systémy, informační vyhledávání, design databází, zpracovávání

přirozeného jazyka a konceptuálních grafů využívá široké spektrum aplikací: od databázových rozhraní, nástrojů vyhledávání textu až po odvozování. Jednotlivá témata budou rozvinuta ve třetí a čtvrté kapitole. Tento fakt jen dokazuje, že teoretický formalismus konceptuálních grafů lze volně ohýbat a následně implementovat do praktických aplikací z nejrůznějších oblastí dle potřeby. Za léta vývoje se ustálily termíny, které označují některé specifické druhy konceptuálních grafů:

- **Základní konceptuální grafy (Basic conceptual graph)**

Tyto konceptuální grafy obsahují pouze uzly konceptů a vztahů a hrany mezi nimi. Koreferenční spojení je možné pouze mezi individuálními koncepty. (Chein, Mugnier, 2009)

- **Jednoduché konceptuální grafy (Simple conceptual graph)**

Tyto grafy na rozdíl od základních umožňují koreferenční spojení mezi jakýmkoliv koncepty, tudíž mohou být spojeny i koncepty, které mají v poli referenta zahrnutí další konceptuální graf atd. (Chein, Mugnier, 2009)

- **Fuzzy konceptuální grafy (Fuzzy conceptual graph)**

Konceptuální grafy využívající fuzzy logiku pro reprezentaci neurčitosti.

- **Rozšířené konceptuální grafy (Extended conceptual graph)**

Sowa tohoto označení užívá pro konceptuální grafy, které pracují s lambda kalkulem, funkcemi, kontexty atd. (Sowa, 1984, 2000, 2008)

## 2 Konceptuální grafy dle John F. Sowy

### 2.1 Definice konceptuálních grafů

Obecně je možno říci, že konceptuální grafy jsou formálním nástrojem pro zaznamenání explicitních znalostí. Zachycují abstraktní, konceptuální strukturu jazykové podoby vyjádřené znalosti. Metamodel konceptuálních grafů, tj. představa o tom, z jakých typů částí a jakým způsobem se mohou konceptuální grafy vytvářet, vychází z lingvistické znalosti gramatické stavby vět přirozeného jazyka. Vychází z pojetí, že věty v přirozeném jazyce vyjadřují vztahy mezi koncepty. (Pavlovská, 2002)

Sowa definoval dvě formální sémantiky pro konceptuální grafy. První z nich je sémantika predikátové logiky definované skrze operátor  $\Phi$ , která mapuje konceptuální grafy do vzorců predikátové logiky. Druhá je sémantika založená na grafech jako takových. v této práci nebudu popisovat mapování z konceptuálních grafů do predikátové logiky, protože to není nezbytné pro účel práce. Více je ale možno se dozvědět například z kapitoly věnující se tomuto tématu v knize Graph-based Knowledge Representation (Chein, Mugnier, 2009, str. 93). Sowa navrhl i syntax a i když není pro konceptuální grafy závazná, tak je popsána v kapitole 2.2, protože je v ní pracováno v celé práci.

#### 2.1.1 Vlastnosti konceptuálních grafů

Konceptuální graf je orientovaný, konečný, souvislý a bipartitní graf. (Sowa, 1984) Souvislý graf je takový graf, pro který platí, že pro všechny dvojice jeho uzlů existuje alespoň jedna cesta, která je spojuje. Konečný graf je takový, který má konečný počet uzlů a hran. Pojmem bipartitní graf se v teorii grafů označuje takový graf, jehož množinu vrcholů je možné rozdělit na dvě disjunktní množiny tak, že žádné dva vrcholy ze stejné množiny nejsou spojeny hranou a hrany vedou pouze z jedné skupiny do druhé, přičemž jedna tato skupina přesně odpovídá konceptům a druhá konceptuálním vztahům. (Sowa, 2000, s. 477) Pokud jde z každého vrcholu jedné množiny hrana do každého vrcholu druhé množiny, mluvíme o úplném bipartitním grafu. (Foltýnek a Dannhoferová, 2011)



### 2.1.2 Formální definice

**John F. Sowa definoval konceptuální graf jako konečný, spojitý a bipartitní graf, pro který platí:**

- a) Konceptuální graf obsahuje dva typy uzlů, které se nazývají **koncepty a konceptuální vztahy** (neboli vztahy mezi koncepty).
- b) Každý konceptuální vztah **v** má jednu či více hran **h**. Každá hrana **h** konceptuálního grafu **G** musí spojovat konceptuální vztah **v** v **g** s konceptem **k** v konceptuálním grafu **G**. Hrana nikdy nespojuje dva koncepty ani dva vztahy, vždy spojuje koncept se vztahem nebo naopak. Hrany mezi těmito uzly jsou orientované. Říkáme, že hrana patří konceptuálnímu vztahu **v** a je připojena ke konceptu **k**.
- c) Pokud má konceptuální vztah **n** hran, je nazýván **n**-adickým a jeho hrany jsou označeny **1,2,...n**. Termín *monadický* je synonymní s 1-adický, *dyadický* s 2-adický a *triadický* s 3-adický.
- d) Jediný koncept sám o sobě může tvořit konceptuální graf. v konceptuálním grafu mohou existovat koncepty, které nejsou spojeny s žádným konceptuálním vztahem, ale každá hrana patřící nějakému konceptuálnímu vztahu musí být spojena s právě jedním konceptem.
- e) Konceptuální graf **G** obsahující **n** konceptuální vztahy může být zkonstruován z **n** hvězdotvých konceptuálních grafů – s jedním pro každý konceptuální vztah **v** v **G**.
- f) Nespojitě hvězdotivé konceptuální grafy se dají spojit do jednoho spojitého konceptuálního grafu překrytím dvou identických typů konceptu. Příkladem je spojení dvou grafů pomocí identického konceptu [JET] do jednoho výsledného grafu:

$[OSOBA : Jan] \leftarrow (AGENT) \leftarrow [JET] \rightarrow (MÍSTO) \rightarrow [MĚSTO : Brno]$

*Vzor. 2.1 Spojení dvou identických typů konceptu do jednoho konceptuálního grafu (Sowa, 1984)*

- g) Jediný koncept sám o sobě může tvořit konceptuální graf. v konceptuálním grafu mohou existovat koncepty, které nejsou spojeny s žádným konceptuálním

vztahem, ale každá hrana patřící nějakému konceptuálnímu vztahu musí být spojena s právě jedním konceptem. (Sowa, 1984)

### 2.1.3 Speciální druhy konceptuálních grafů

Sowa přiřazuje speciální jména těmto 3 druhům konceptuálních grafů (1984) :

1. **blank** (prázdný) je prázdný konceptuální graf, kde není žádný koncept, konceptuální vztah nebo hrana
2. **singleton** (osamělý) je samostatně stojící koncept, který není spojen s žádným konceptuálním vztahem
3. **star** (hvězda) je konceptuální graf, který sestává z jediného konceptuálního vztahu a připojených konceptů. (Dvě i více hran konceptuálního vztahu může být připojeno k jednomu konceptuálnímu vztahu) (Sowa, 1978)

## 2.2 Notace konceptuálních grafů (ISO/IEC 24707, 2007)

Neformálně je konceptuální graf struktura dvou typů uzlů: konceptů a konceptuálních vztahů, kdy každá hrana spojuje uzel konceptu s uzlem konceptuálního vztahu. Konceptuální grafy využívají dva základní typy notace. Mohou být znázorněny graficky ve formě diagramu či zapsány v podobě lineárního textového kódu. Grafická notace je většinou lépe čitelná a názorná a lineární notace zase zabírá méně prostoru. Avšak obě formy mohou znázorňovat stejný abstraktní graf a mohou být automaticky přeloženy do jiných verzí logiky (Sowa, 2000, str. 477)

V diagramu jsou **koncepty zobrazovány jako obdélníky, konceptuální vztahy jako kruhy a hrany jsou šipky**, které spojují obdélníky s kruhy. v podobě lineárního textového kódu jsou koncepty ohraničeny hranatými závorkami a koncepty kulatými závorkami.

Je však třeba zdůraznit, že označovací soustava obdélníků a kruhů je sice vyhovující a pohodlná, avšak není závazná. Podstatné je to, že abstraktní syntax specifikuje konceptuální grafy jako matematickou strukturu, jejíž matematické základy jsou dostatečně obecné na to, aby mohly reprezentovat jakýkoliv soubor vztahů mezi diskrétními entitami bez závazku k jakékoliv konkrétní notaci či implementaci. (Sowa, 2001)

Oba typy notace, jak grafická, tak lineární textová, jsou navrženy pro komunikaci mezi lidmi nebo mezi člověkem a počítačem. i přesto, že se jednotlivé způsoby notace zdají velmi odlišné, mají stejnou sémantiku a stejné logické základy. Jakákoliv sémantická informace vyjádřená v každém z nich může být přeložena do jiné notace bez ztráty nebo zkreslení informace.

### 2.2.1 Předpoklady validní konstrukce konceptuálního grafu:

Sám Sowa ve své knize (Sowa, 1984 s. 73) deklaruje, že byt' preferuje notaci pomocí diagramů, protože ty do značné míry pomáhají lidem s vizualizací vztahů, tak je jeho teorie konceptuálních grafů absolutně nezávislá na způsobu, jakým je graf zobrazován i na notaci. Sowa však uvádí tři předpoklady, které jsou dle něho nezbytné k tomu, aby mohl být konceptuální graf vůbec vyprojektován.

#### 1. Koncepty jsou diskrétní (nespojité) jednotky.

V oboru umělé inteligence je výraz koncept užíván pro uzly, které kódují informace v grafech a sítích, koncept je tedy základní jednotkou pro reprezentaci znalostí. Už sama definice konceptu jako „jednotky“ předpokládá, že je koncept diskrétní povahy. Tento předpoklad je podpořen i faktem, že diskrétní vztahy jsou lépe a přesněji zapamatovatelné, než-li spojité kvantify.

Pokud mají lidé popsat nebo nakreslit nějakou scénu ze své paměti, tak si většinou vybaví diskrétní vlastnosti a velikosti, čas i teplotu si většinou zapamatují pomocí porovnávání: „Byl vysoký jako já.“ „Čekala jsem, dokud se nesetmělo.“ „Voda byla jako led.“ Ve skutečnosti jsou však kvality (jako je teplota, vzdálenost, emocionální stav) spojité a to i když jsou v přirozeném jazyce popisovány pomocí diskrétních slov typu: studený, teplý, horký, šťastný, smutný, daleko, blízko atd. Všechny známé lidské jazyky také umějí pojmenovat pouze diskrétní množinu barev z celého existujícího spojitého barevného spektra.

Pro přizpůsobení diskrétních slov, aby více korespondovaly se spojitým světem, využívají přirozené jazyky tzv. fuzzy slov, jako jsou téměř, radši, víceméně, přibližně, skoro, ne zcela atd. Tato slova však jen rozšiřují a škálují existující diskrétní kvantify: skoro horké je jen další diskrétní stav. Tuto situaci řeší fuzzy logika, která takovými termínům přiděluje určité hodnoty. Avšak je třeba zdůraznit, že fuzzy kalkulus je

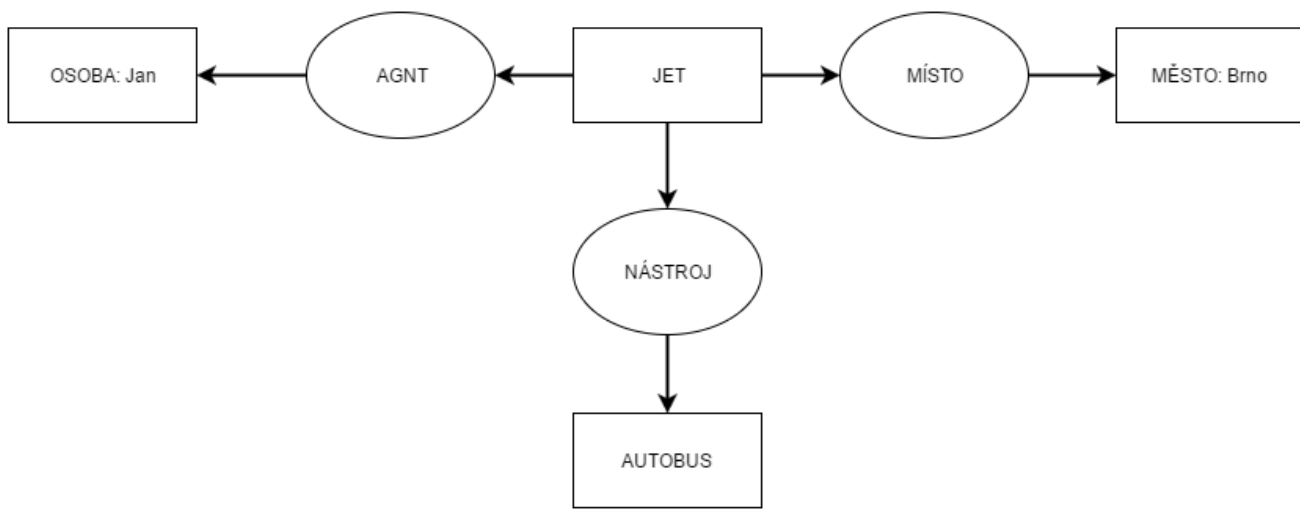
schopen rozlišovat na tak detailní úroveň, jak žádný přirozený jazyk nedokáže. Lidský mozek tedy sice využívá spojitých i nespojitých procesů, avšak konceptuální grafy zachycují pouze koncepty jako diskrétní jednotky.

**2. Kombinace konceptů nejsou difúzní směsi, ale uspořádané struktury.**

**3. V konceptech jsou zaznamenány pouze diskrétní vztahy.**

Spojité formy musí být přiblíženy schématem diskrétních jednotek. (Sowa, 1984)

### 2.2.2 Grafická forma



Obr. 2.1 Grafická forma grafu zobrazující větu „Jan jede do Brna autobusem.“(Sowa, 1984)

Koncepty v grafu na Obr. 2.1 jsou reprezentovány obdélníky: [JET], [OSOBA: Jan], [MĚSTO: Brno], [AUTOBUS]. Konceptuální vztahy jsou reprezentovány kruhy či ovály: (AGENT) vztahuje koncept [JET] ke konceptu [OSOBA: Jan]. Tento konceptuální graf obsahuje 4 koncepty: [JET], [OSOBA: Jan], [MĚSTO: Brno], [AUTOBUS]. Dále obsahuje 3 konceptuální vztahy: (AGENT), (MÍSTO) a (NÁSTROJ) a tyto koncepty a vztahy spojuje 6 hran. v grafu není žádná hrana, která by spojovala koncept s konceptem nebo vztah s vztahem. Dvě hrany v grafu patří ke vztahu (AGENT), dvě ke vztahu (MÍSTO) a dvě ke vztahu (NÁSTROJ).

#### Čtení grafu:

U n-adických vztahů jsou hrany číslovány od 1 do  $n$ . u dyadických vztahů může být číslo 1 nahrazeno šipkou směrem ke vztahu a číslo 2 šipkou mířící pryč od vztahu. Jako mnemotechnickou pomůcku můžeme číst šipku mířící ke kruhu jako: „Má“ a šipku mířící od

kruhu jako: „*kterým/ou je.....*“ Zjistila jsem, že byt' je tato mnemotechnická pomůcka vymyšlena původně pro anglický jazyk, lze ji aplikovat i na jazyk český.

- JET má agenta, kterým je osoba Jan
- JET má místo, kterým je město Brno
- JET má nástroj, kterým je Autobus

### 2.2.3 Forma lineárního textového kódu

#### Základní forma lineárního textového kódu:

V této formě jsou koncepty znázorněny hranatými závorkami, konceptuální vztahy kulatými závorkami. Pomlčka na konci řádku znamená, že k tomuto konceptu jsou připojeny i další vztahy uvedené na následujících řádcích.

[JET] –

(AGENT) → [OSOBA: Jan]

(MÍSTO) → [MĚSTO: Brno]

(NÁSTROJ) → [AUTOBUS].

*Vzor. 2.2 Lineární forma zápisu věty: „Jan jede do Brna“*

## 2.3 Ontologie v konceptuálních grafech

### 2.3.1 Ontologie a její význam

Termín ontologie je převzat do informační a počítačové vědy z filosofie, v které představuje disciplínu, která se zabývá nejjobecnějšími otázkami jsoucna a bytí, existence a reality (Mařík, Štěpánková a Lažanský, 2013). Za zakladatele ontologie je považován antický filosof Permenidés (5. století př.n.l.). První ontologie se objevily se snahami antických před Sokratovských myslitelů předložit ontologické koncepce mýtického výkladu světa (Kratochvíl a Bouzek, 1994), kdy se snažili redukovat veškerou skutečnost na jediný princip. Pohled na ontologii se v průběhu dějin přirozeně měnil v závislosti na proměnách a vývoji vnímání světa a jeho výkladu. (Černík, 2000, str. 93)

Mnoho autorů včetně Sowy (1984) přikládá zásadní význam proměně v pojetí studia ontologií, kterou sebou přinesla platónská a především aristotelovská škola. Aristoteles se ve svém díle *Kategorie* snažil rozčlenit veškeré skutečnosti a identifikovat u nich základní vlastnosti: jeho čtyři třídy a deset kategorií mělo pokrýt veškeré jsoucní. Co je však nejzajímavější, že jeho kategorie připouštějí hierarchickou strukturu, což je vlastní dnešnímu pohledu na termín ontologie. (Vlček, Vacura, 2014) Samotný termín Ontologie se prvně objevil až v 17. století v pracích německých filosofů R.Gockela a J. Lorharda (Mařík, Štěpánková a Lažanský, 2013).

Jedním z prvních, kdo poukazyval na nutnost implementace ontologií do počítačové a informační vědy byl právě John F. Sowa ve svém díle *Conceptual Structures* (1984). o několik let později Sowa (2000, str. 492) definuje ontologii následovně: *„Předmětem ontologie je studium kategorií věcí, které existují nebo mohou existovat v nějaké oblasti. Výsledkem takového studia, zvaného ontologie, je katalog typů věcí považované za existující v oblasti zájmu D z pohledu osoby, která používá jazyk L pro účely popisu oblasti D.“*

Starší definice ontologie v oblasti informatiky tvrdí, že ontologie je explicitní specifikace konceptualizace (Gruber, 1993) nebo je možno říci explicitně popsaná sdílená konceptualizace (Borst, 1997). Konceptualizace je proces vedoucí k vytvoření konceptuálního modelu a v souvislosti s pojetím Grubera a Borsta (1993, 1997) je požadavek na konceptualizaci dán faktem, že ontologie zachycuje strukturu oblasti zájmu a tedy znalost o celé oblasti zájmu nikoliv o konkrétních stavech věcí v ní. Specifikací je myšlen fakt, že je konceptualizace specifikována či explicitně popsána užitím konkrétního modelovacího jazyka a konkrétních termů. Tato specifikace má často formální charakter, což je absolutně nezbytné k automatizaci procesů a práci s ontologiemi. (Obitko, 2007)

Přenesení pojmu ontologie do oblasti počítačové a informační vědy není náhodné. Již první informační systémy sebou přenášely nějaký model či zobrazení vybrané části reality, která byla v případě těchto prvních systémů dobře vymezena a systém pracoval s pouze omezenou množinou předem vymezených a dobře známých objektů. Avšak s rozvojem informačních technologií a především s neskutečně velkým nárůstem informací dostupných v elektronické podobě sebou přináší snahu zakomponovat do informačních systémů stále větší a komplexněji zpracovnou část reality. Objevení internetu, který obsahuje ze

sémantického hlediska neomezenou množinu informací tento požadavek ještě umocňuje. Jako důsledek toho byl formulován požadavek na vytvoření sémantického webu, který počítá s existencí nezávislého referenčního rámce, který by umožnil reprezentovat sémanticky odlišné informace, explicitně popsat jejich vzájemný vztah a také umožnil vzájemnou interakci mezi různými informačními systémy zpracovávající sémanticky odlišné informace.

Právě ontologie může sloužit jako nástroj pro reprezentaci a sdílení znalostí obsažených v konceptualizaci. Ontologie musí charakterizovat třídy (v této práci nazývané typy) objektů jejich vlastnostmi, nikoliv vyjmenováním jednotlivých instancí, jinak ji není možné využít pro odvozování. Díky tomuto pravidlu lze na základě definičních vlastností jednotlivých tříd lze do těchto tříd zařazovat další jednotlivé instance. v pojetí reprezentace znalostí a odvozování tedy ontologie stojí nad znalostní bází, kterou uspořádává do struktury. (Mařík, Štěpánková a Lažanský, 2013; Sowa, 2000; Obitko, 2007).

### 2.3.2 Sowaův návrh ontologie nejvyšší úrovně

Sowa (2000, str. 84) tvrdí, že „...neinterpretovaná logika (například predikátový kalkulus, KIF, konceptuální grafy) je ontologicky neutrální ve smyslu toho, že je schopna vyjádřit všechny možné vztahy mezi entitami.“ Což by se dalo vysvětlit tak, že konceptuální grafy samy o sobě nenesou žádnou informaci, jsou pouhým prázdným formátem. Ale kombinací konceptuálních grafů a ontologie vznikne jazyk schopný vyjádřit vztahy mezi entitami určené oblasti zájmu. Konceptuální grafy lze tedy využít jako formalismus při snaze zachytit ontologii určité oblasti nebo domény.

Neformální ontologie může být specifikována například katalogem typů, které jsou definované jen výroky v přirozeném jazyce. Avšak v této práci se přikláním k pojetí ontologie jako tzv. formální nebo deskriptivní ontologie, která zkoumá otázky klasifikační povahy: neboli jaké třídy (v tomto textu označené jako typy) entit (jsoucen) existují a jaké jsou mezi nimi typy vztahů. Toto pojetí také předpokládá existenci určitého sdíleného univerzálního konceptuálního schématu, které sdílejí všichni lidé (u všeobecné ontologie) nebo určitá komunita (u ontologie reprezentující dílčí oblast zájmu dané komunity) (Strawson, 1990). Formální ontologie je v pojetí Sowy (2000) specifikována částečně uspořádanými množinami návěští (jmen) typů konceptů a typů vztahů a vztahů mezi nimi.

## **Typy konceptů a vztahů v ontologii**

Neformálně řečeno je typ návěštím (jménem), které je přiřazeno množině entit podobného pojetí a vlastností. Formálně je typ specifikací množiny nebo kolekce entit, které existují nebo by mohly existovat v nějaké oblasti diskurzu (Sowa, 2000, str. 98). Přestože typ je jméno skupiny entit, vždy existuje pouze v abstraktní rovině. Avšak neplatí, že cokoliv abstraktního je typem. Např. „2“ je abstraktní pojem, ale není typem, nýbrž instancí typu „PŘIROZENÉČÍSLO“. (Lukášová, 2010) Typ se skládá z pojmenování typu, které může být jakékoliv (například MĚSTO) a z definice, která charakterizuje o jaký typ se jedná (například: Geograficky specifikovaný útvar, pro který je typická kompaktnost a koncentrace zástavby a vysoká hustota osídlení).

### **Definice typu:**

Typy lze definovat nejrůznějšími způsoby, ale základní jsou čtyři operace jsou:

- **Extenze:**

Extenze definuje typ vyjmenování prvků množiny, kterou typ definuje. Tento přístup je vhodný jen v případě výskytu malého počtu takových individuí. Příkladem je definice typu ČlenRodinyČermákovi, kdy stačí vyjmenovat seznam <Petr, Michaela, Hana, Filip>. (Petersen, 2005)

- **Intenze:**

Intenze definuje typ vyjmenováním vlastností, které definují individuum patřící do daného typu. Intenze je také definování pravidel pro rozhodování, zda je individuum instancí daného typu či nikoliv. Tento postup je praktický při formulaci typů obsahujících velké množství individuí, ale je k tomu zapotřebí použít nějaký jazyk pro definici pravidel. Například typ SAVEC může být definován intenzí za pomoci pravidla: „Savec je Zvíře, které je pohlavní, teplokrevné, mláďata mají srst a jsou krmena z mléčných žláz. Intenze pravidlem lze například uplatnit pro typ „LICHÉ\_ČÍSLO“: Liché číslo nelze beze zbytku dělit dvěma.“ (Sowa, 2000, str. 95; Petersen, 2005)



- **Axiomy:**

Axiom je pojem z matematické teorie, který označuje nějaký výrok, který není nutné dokazovat a u kterého se automaticky předpokládá, že je pravdivý. Tímto způsobem Sowa nadefinoval primitiva jeho ontologie. (Sowa, 2000, str. 412)

- **Odkazem na jiné typy s diferenciací (přidáním odlišnosti):**

Tento způsob definice typů je druhem intenze často používaným, protože je praktický. Pokud existuje definovaný typ STUDENT je pak jednoduché definovat typ „STUDENT\_UISKU“ následovně: STUDENT\_UISKU je STUDENT s diferenciací, že on nebo ona byl/a přijat/a na UISK a studují tam.“ Tato metoda se používá při definování hierarchie typů konceptů a to definováním nových typů za pomoci konceptuálních grafů, což umožňuje využít již vytvořené konceptuální grafy. (Petersen, 2005)

Aby bylo možno znázorňovat různé úrovně všeobecnosti, jsou typy konceptu a vztahů částečně uspořádány do hierarchie typů, kdy je zaveden vztah podtyp-nadtyp. Hierarchie typů je tedy částečně uspořádaná množina  $T$  všech  $t$ , jejíž prvky se nazývají *návěští typu*. Částečného uspořádání množiny  $T \{t_1, \dots, t_n\}$  je dosaženo pomocí vztahu podtyp se symboly  $\leq$  pro podtyp,  $<$  pro řádný podtyp,  $\geq$  pro nadtyp a  $>$  pro řádný nadtyp.  $t_1 \leq t_2$  znamená, že  $t_1$  je specializací  $t_2$  a  $t_2$  je zobecněním  $t_1$  neboli  $t_2$  zahrnuje  $t_1$  a každá entita typu  $t_1$  je také typem  $t_2$ . (Sowa, 2001; Chein a Mugnier, 1992, 2009) Jinak řečeno typ SLON je podtypem a specializací typu ZVÍŘE. Specializace (v tomto případě SLON) má všechny vlastnosti zobecnění ZVÍŘE a navíc přidává některá další omezení nebo je možno říci specializované charakteristiky a vlastnosti (Petersen, 2005).

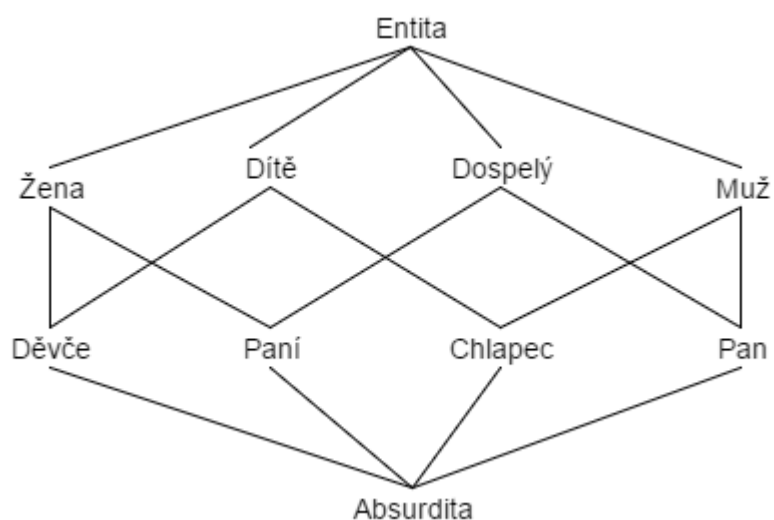
Každé návěští typu  $t$  z  $T$  je specifikováno jako primitivní nebo definované. Pro každý koncept je typem konceptu buď výše popsané návěští z množiny  $T$  (*primitivní návěští*), nebo monadický lambda výraz (*definované návěští*) (viz kapitola 2.4). To v praxi znamená, že v hierarchii typů není vždy nutné vyjmenovat všechny možné typy, ale mohou být nadefinovány monadickým lambda výrazem. (Sowa, 2001)

### Druhy návěští typu:

- **Primitivní návěští** jsou pouhými jmény zařaditelnými do hierarchie **T**. Jinak řečeno se jedná o návěští typu, která se neobjevila v definici typu. Každá hierarchie typů obsahuje dvě primitivní typová návěští – entitu a absurditu (Sowa, 2001).
- **Entita** neboli univerzální typ (**T**) je typ, který je nadtypem všech ostatních typů v hierarchii a reprezentuje libovolnou entitu v množině zařazenou.
- **Absurdita** neboli absurdní typ (**⊥**) je typem, který je podtypem všech ostatních typů v hierarchii. Je možno říci, že v hierarchii není nic níže nežli absurdita. (Petersen, 2005)
- Pokud je **t** nějakým návěštím typu, tak je  $T \geq t$  a  $t \geq \perp$  a především  $T \geq \perp$ .
- **Definovaná návěští**. Pro každé definované návěští typu existuje monadický lambda výraz, který se nazývá jeho definicí. Definovaná návěští jsou tedy definována pomocí monadických lambda výrazů.

### Dědičnost

Dědičnost se týká vlastností entit a vyplývá z částečného uspořádání množiny **T**  $\{t_1, \dots, t_n\}$  se vztahem podtyp – nadtyp. Pokud platí, že  $t_1 \leq t_2$ , tak  $t_1$  tzv. dědí neboli má všechny vlastnosti  $t_2$ . Nebo jinak řečeno, cokoliv, co platí o  $t_2$  platí i o  $t_1$ . Například  $SAVEC \leq ZVÍŘE$  znamená, že jakékoliv tvrzení je pravdivé o jakékoliv instanci typu **ZVÍŘE** je pravdivé i pro jakoukoliv instanci typu **SAVEC**. Opačně to ovšem neplatí, protože například fakt, že savci krmí svá mláďata mlékem se nedá říci o všech zvířatech. Speciálním typem je vícečetná dědičnost, kdy jeden podtyp může mít vlastnosti dvou nadtypů. k tomu například dochází při konjunkci primitiv, kdy vytvořené podtypy mají vlastnosti definované axiomou obou primitiv. (Petersen, 2005)

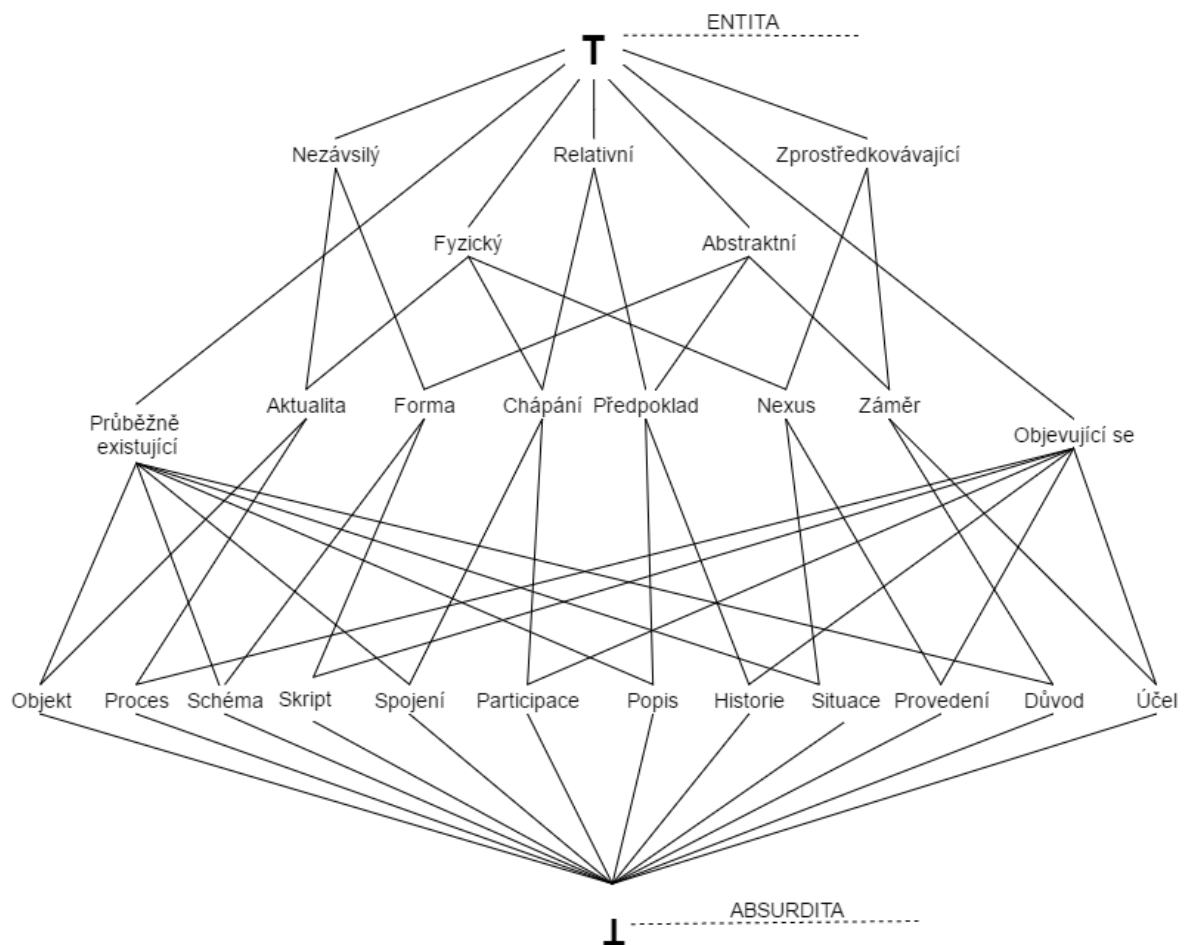


Obr. 2.2 Svaz reprezentující vícečetnou dědičnost

### Obecná ontologie

Sowa navrhl univerzální strukturu ontologie, kterou reprezentoval ve formě matice a Hasseova diagramu neboli svazu<sup>12</sup>, který se používá pro zobrazování konečných částečných množin (Bartl, 2006 str. 16).

<sup>12</sup> Uspořádaná množina, v níž ke každým dvěma prvkům existuje supremum i infimum, se nazývá svaz. (Kučera, 2010 str. 5)



Obr. 2.3 Svaz typů reprezentující obecnou ontologii Johna F. Sowy (Sowa, 2000, str. 72)

	Fyzický		Abstraktní	
	Průběžně existující	Objevující se	Průběžně existující	Objevující se
Nezávislý	Objekt	Proces	Schéma	Skript
Relativní	Spojení	Participace	Popis	Historie
Zprostředkovávající	Provedení	Situace	Důvod	Účel

Obr. 2.4 Matice zobrazující dvanáct typů nacházejících se v centru ontologie zobrazené na Obr. 2.3 (Sowa, 2000 str. 75)

Jak je vidět na Obr. 2.3, svaz představující Sowovu ontologii obsahuje univerzální typ označený symbolem T a nazvaným Entita, kdy všechny ostatní typy jsou podtypy této Entity. Na opačném konci ontologie je symbol  $\perp$ , který se nazývá Absurdita, kdy žádný objekt není instancí typu Absurdita a všechny ostatní typy jsou nadtypy Absurdity. Některé typy jsou popsány výrazem označující vlastnost (například FYZICKÝ nebo ABSTRAKTNÍ). v takovémto

případě je nutno chápat výraz jako vlastnost pojmu entity (tedy například fyzická entita). (Mařík, Štěpánková a Lažanský, 2013)

Sedm základních typů nejvyšší úrovně: FYZICKÝ NEBO ABSTRAKTNÍ, NEZÁVISLÝ, RELATIVNÍ NEBO ZPROSTŘEDKOVÁVAJÍCÍ, PRŮBĚŽNĚ EXISTUJÍCÍ nebo OBJEVUJÍCÍ SE jsou tzv. primitivními typy neboli primitiva. Primitivum nemůže být definováno pomocí jiného typu stejné ontologie. Význam primitiv je určen axiomy nebo je možno říci, že mají definovanou sémantiku pomocí logických výrazů. Typy na nižších úrovních ontologie jsou vytvářeny za pomoci konjunkcí primitiv. Sowa sémantiku primitiv své ontologie definoval v knize Knowledge Representation (2000), kde také podrobně vysvětluje i ostatní typy na nižších úrovních své ontologie. Pro potřeby této práce však postačí uvést několik ilustrativních příkladů:

- Primitivní typ ZPROSTŘEDKOVÁVAJÍCÍ je entita, která přivádí ostatní entity do vztahu.
- Primitivní typ FYZICKÝ je entita, která má lokaci v prostoru a čase, což se dá vyjádřit za pomoci axiomů:
  - Cokoliv fyzického je lokalizováno v nějakém prostoru:  $(\forall x: \text{Fyzický})(\exists y: \text{Místo})\text{loc}(xy)$ .
  - Cokoliv fyzického se objevuje v nějakém bodě v čase:  $(\forall x: \text{Fyzický})(\exists t: \text{Čas})b\text{Čas}(xt)$ .
- Typ OBJEKT vznikne konjunkcí typů FYZICKÝ, NEZÁVISLÝ a PRŮBĚŽNĚ EXISTUJÍCÍ.
- Typ NEXUS je například fyzická entita zprostředkovávající (dává do vztahu) dvě nebo více jiných entit. Příkladem takového typu může být třeba typ vztahu (AGENT). (Sowa, 2000, str. 500)

### 2.3.3 Koncept

Definovat pojem „koncept“ není snadné. Pojem pochází z původně latinského *conceptio* – „početí, pojetí, shrnutí“ či *conceptus* – „sebrání, početí“, významově většinou představuje „prvotní náhled, abstrakt“ náskok, osnovu, skicu, rozvrh. Je také synonymem slovu koncepce, které v pojetí filosofie znamená pojem, představa, myšlenka, rozumové uchopení či pojetí tématu v logické sémantice. (Holub, Lye, 1992). Pro tuto práci postačí říci, že koncept může být cokoliv konkrétního či abstraktního (osoba, akce, vlastnost, matematická funkce, čas, věk atd.), co někdo pokládá za entitu. Sám Sowa tvrdí, že uzly

konceptů v jeho konceptuálních grafech představují: entity, atributy, stavy a události (Sowa, 1984). Tato kapitola rozebírá, jak je pojem „koncept“ uchopen v teorii konceptuálních grafů. Jaký je způsob jeho notace a jaké má nadefinované vlastnosti.

V konceptuálních grafech je koncept zakreslen v kruhu nebo v kulaté závorce a většinou bývá spojen hranou s nějakým konceptuálním vztahem. Dvěma konceptům připojením k jednomu vztahu se říká „*sousedské koncepty*“ (Cao, 2010). v teorii konceptuálních grafů se každý koncept skládá ze dvou složek: **typu konceptu t** a **referent r**. Není jasně specifikováno, jak jsou typ a referent v grafu reprezentovány a jejich zápis se většinou liší dle způsobu zápisu grafu. Například v databázi jsou často vyjádřeny pomocí dvou návěští, kdy jedno návěští specifikuje typ konceptu a druhé referent. v podobě lineárního textového kódu nebo i v diagramu je většinou typ zapsán vlevo a referent vpravo a jsou od sebe odděleny dvojtečkou: [TYP : REFERENT]. (Sowa, 2001)

Jestliže referent chybí (je prázdný), je jím implicitně existenční kvantifikátor. v konceptu na Obr. 2.5 [AUTOBUS], je „Autobus“ typem a pole referentu je prázdné, což značí existenční kvantifikátor, který říká, že existuje nějaký autobus bez bližšího určení. v druhém konceptu [OSOBA: Jan], je typ „Osoba“ a referent je „Jan“, tzn. graf obsahuje tvrzení: „existuje osoba a ta se jmenuje Jan“. Typ konceptu nemůže zůstat na rozdíl od referentu prázdný, např. [ : Jan] nemůže nikdy nastat.



Obr. 2.5 Koncept obsahující referent a koncept bez referentu

#### 2.3.4 Typ konceptu

Obecně lze říci, že typ konceptu je návěští (jméno), které je přiřazeno množině entit podobného pojetí. Některé typy konceptu jsou všeobecnější nežli jiné (Například: Osoba>Zaměstnanec>Manažer). Aby bylo možno znázorňovat různé úrovně všeobecnosti, jsou typy konceptu částečně uspořádány do hierarchie typů, kdy je zaveden vztah podtyp-nadtyp, jak bylo popsáno v kapitole 2.3.2. Hierarchie typů je tedy částečně uspořádaná množina **T** všech **t**, jejíž prvky se nazývají *návěští typu konceptu*. Částečné uspořádání návěští

typu musí být v souladu s pravidly odvození definovanými nad lambda výrazy. (Sowa, 2001; Moreau, Leclère, Chein a Gutierrez, 2007)

### Referent konceptu

Referent konceptu ukazuje na výskyt či výskyty (instance) z typu konceptu, které jsou pojmovým obsahem konceptu (ISO/IEC 24707, 2007). Referent konceptu je specifikován **kvantifikátorem a designátorem**. Referent konceptu determinuje entitu nebo soubor entit na které koncept odkazuje. Designátor specifikuje referent tím, že definuje jeho formu (*literál*), tím že na něj ukazuje (*lokátor*) nebo tím, že ho popisuje (*deskriptor*). Existenciální kvantifikátor deklaruje, že alespoň jedna instance daného typu existuje: definovaný kvantifikátor může specifikovat množství. (Sowa, 1984, 2000)

### Kvantifikátor

- 1) **Existenciální kvantifikátor** je reprezentován buď prázdným referentem nebo symbolem  $\exists$ . Referent, jehož kvantifikátor  $q$  je existenciální, se nazývá existenciálním referentem.

Například lineární zápis [KOČKA]  $\rightarrow$  (NA)  $\rightarrow$ [MATRACE]. (pro větu „Existuje nějaká kočka, která je na nějaké existující matraci.“).

- 2) **Definovaný kvantifikátor** je symbol nebo výraz v rozšířené syntaxi, který může být přenesen do konceptuálních grafů, jež obsahují pouze existenciální kvantifikátory. Definované kvantifikátory se dále dělí na:

- **univerzální kvantifikátor  $\forall$ ,**

[ŽIJÍCÍRYBA :  $\forall$ ]  $\rightarrow$  (ATTR)  $\rightarrow$  [MOKRÝ]. “Všechny žijící ryby jsou mokré“

- **množství** (např. @1, @2 atd.)

- **množina výčtem** (např. {Tomáš, Petr, Pavel}, kde Tomáš, Petr a Pavel jsou designátory)

## Designátor

Designátor specifikuje referent buď poukázáním na jeho formu (*literál*), ukázáním na umístění referentu (*lokátor*), tím že referent popíše pomocí jiného konceptuálního grafu (*deskriptor*) a konečně nedefinovaný designátor, který o referentu nic nevyovídá.

1) **Literál** je syntaktická reprezentace formy referentu. Jde tedy o zakódovaný záznam referentu, který obsahuje informace o použitém způsobu kódování. v konceptuálních grafech je možno identifikovat tři typy literálů:

- **číslo [Číslo : 18]**

Musíme jej odlišovat od kvantifikátoru **@18**, který udává počet, zatímco tady je to číslo 18.

- **řetězec znaků (string) - [STRING : abcd]**

Například v multimediálních systémech může literál tohoto typu kódovat vzorec reprezentující zvuk, grafiku nebo video.

- **kódovaný literál**

Je vždy uvozen znakem % a následně specifikován identifikátorem a řetězcem znaků. Koncept [SITUACE: %Angličtina"A plumber is carrying a pipe." ] nám například poukazuje na situaci, která je popsána anglickým jazykem. Identifikátorem mohou být také formáty jako GIF, AVI atd. :[FOTOGRAFIE: %JPG"krajinka.jpg"]

2) **Lokátor** je symbol, který určuje, jakým způsobem může být referent nalezen. Tento referent tedy říká, kde lze nalézt individuum, k němuž směřuje reference, a to ve fyzickém světě nebo katalogu individuí. v konceptuálních grafech jsou rozlišovány tři druhy lokátorů:

- **Individuální označovače** specifikují unikátní koncept v katalogu individuí například ve znalostní bázi. Tento soubor jmen užívaný pro označení specifických entit je předem vytvořený a je možné ho považovat za primitivní ontologii (Moreau, Leclère, Chein a Gutierrez, 2007)

Např.: [ZAMĚSTNANEC: #591912/7895]



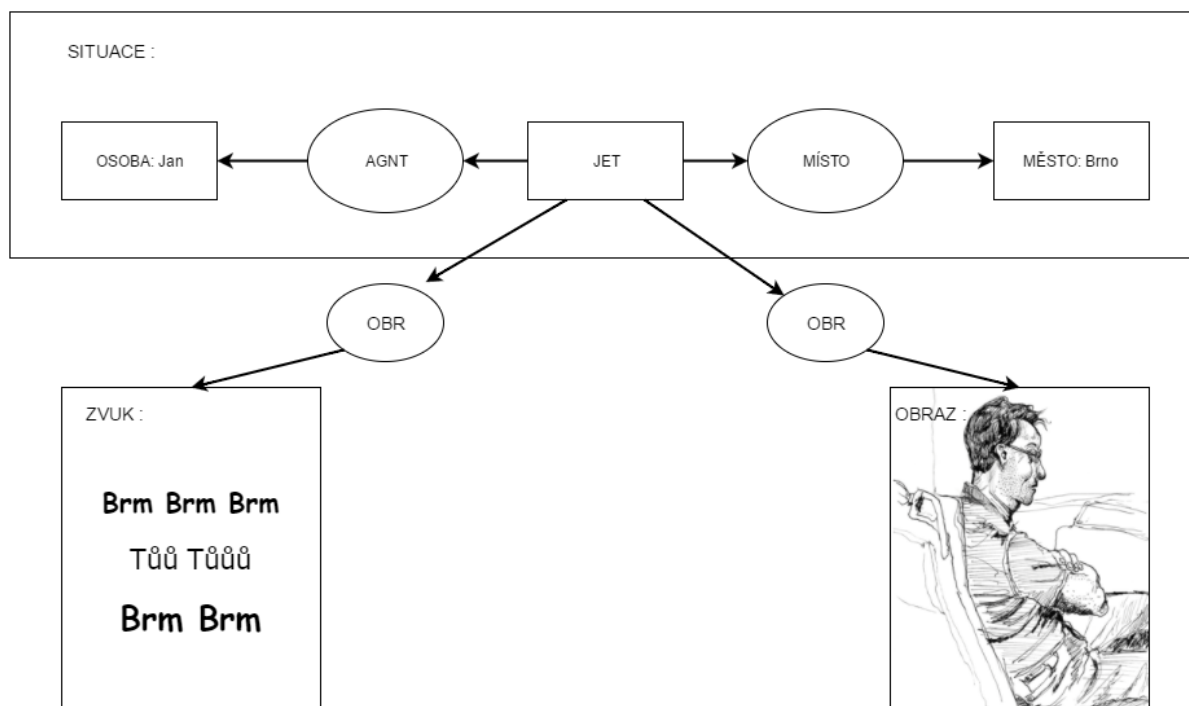
- **Indexační** lokátory jsou používány v situacích, kdy je třeba referovat na individua, jejichž identita závisí na kontextu a může se v závislosti na kontextu měnit.  
Např.: [OSOBA : #Ty], kdy se význam konceptu „osoba“ mění podle kontextu.
- **Jmenné lokátory** jsou symboly, které určují referent pomocí konvencí, jež jsou nezávislé na daném modulu nebo znalostní bázi. Např.: „ISBN-0-534-94965-7“, kdy je mezinárodní standardní číslo knihy definováno nezávisle na grafu, IT systému nebo modulu a je to jasný jednoznačný neopakovatelný identifikátor, u kterého máme jistotu, že představuje pouze jedinou entitu.

3) **Deskriptor** je používán u referentu, kdy je zapotřebí složitějšího popisu za použití jiného konceptuálního grafu. Často k tomu dochází u konceptu typu „SITUACE“ nebo „TVRZENÍ“. Deskriptor je tedy konceptuální graf, který je zahrnut (viz kapitola 2.5) v části referentu daného konceptu. Tento zahrnutý konceptuální graf nějakým způsobem popisuje referent.

[TVRZENÍ: [KOČKA] ← (AGENT) ← [HONIT] → (THME) → [MYŠ]].

*Vzor. 2.3 „Existuje tvrzení, které říká, že kočka honí myš.“*

Typ konceptu je "TVRZENÍ" a referent konceptu je konceptuální graf: [KOČKA]←(AGENT)←[HONIT]→(THME)→[MYŠ]. Koncept s prázdným referentem jako jev tomto případě [KOČKA] má implicitní existenciální kvantifikátor a prázdný konceptuální graf v poli deskriptoru. Jelikož prázdný graf neříká nic o referentu, koncept [KOČKA] sám o sobě znamená „existuje kočka“.



Obr. 2.6 Konceptuální graf s deskriptorem a dvěma literárními referenty (Sowa, 1984)

Pro ilustraci uvádím konceptuální graf na Obr. 2.6 s deskriptorem a dvěma literárními referenty: Jedná se o koncept [SITUACE ] s deskriptorem v podobě zahrnutého konceptuálního grafu, který by mohl být čtený jako: „Jan jede do Brna.“ Uzel konceptu na Obr. 2.6 [SITUACE ] je spojen skrz obrazový vztah (OBR) s konceptem typu „OBRÁZEK“, jehož pole referentu obsahuje zakódovaný literál obrázku, který ilustruje situaci. v notaci CGIF by pole referentu bylo uvozeno znakem „%“, jenž značí kódovaný literál a následoval by identifikátor, jenž by specifikoval formát a řetězec reprezentující zakódování obrázku. Uzel konceptu [SITUACE ] je také propojen skrz vztah (OBR) s konceptem [ZVUK ], jehož referenční pole obsahuje zakódovaný literál reprezentující asociovaný zvuk. v tomto diagramu je grafické znázornění zvuku, standardně jsou však zakódované literály reprezentovány identifikátorem pro typ kódování jako je např. Wav pro zvuk nebo JPEG pro obraz, po kterých následuje řetězec obsahující zakódovaná data. Avšak nějaký multimediální systém by mohl rovnou reprodukovat zvuk ve chvíli, kdy uživatel klikne na uzel. (Sowa, 2000, 2001)

### 2.3.5 Druhy konceptů

Sowa sice navrhl dělit koncepty dle druhu jejich referentu, ke kterému navrhl celou typologii, avšak v praxi se vědecká komunita autorů přiklonila k dělení konceptů pouze na generické a individuální. Dle tohoto pojetí referent identifikuje individuum jako známé (konstanta) a tomu náleží označení individuální koncept, nebo neznámé (proměnná nebo symbol \*) a tomu náleží označení generický koncept.

Existuje množina  $I = \{\#1, \#2, \#3 \dots\}$ , jejíž prvky se nazývají individuální označovače. Funkce referentu může být aplikována na jakýkoliv koncept  $c$ :

- Referent ( $c$ ) je buď individuálním označovačem z množiny  $I$  nebo generickým označovačem  $*$ .
- Když je referent ( $c$ ) z množiny  $I$ , tak je  $c$  nazýván individuálním konceptem
- Když je referent ( $c$ )  $*$  tak je  $C$  nazýván generickým konceptem. (Sowa, 1984, str. 85)

#### Generické koncepty

Generické koncepty tedy zastupuje individua, jejichž identity jsou neznámé. Mohou být existenčně nebo univerzálně kvantifikovány. Generické koncepty ( jak existenčně tak univerzálně kvantifikované) mají v poli typu běžné slovo nebo generický term, které specifikuje nějaké atributy identity o které pojednává, ale platí stejně pro jakoukoliv entitu, která má tyto atributy. Pokud není řečeno jinak, je generický koncept kvantifikován existenčně. Takový koncept je ekvivalentem konceptu s existenciálním kvantifikátorem v poli referentu a buď je pole referentu prázdné [OSOBA] nebo se značí hvězdičkou [OSOBA: \*]. Univerzálně kvantifikovaný koncept musí mít v poli referenta univerzální kvantifikátor: [ŽijícíRyba :  $\forall$ ]. Využití generických konceptů je při reprezentaci znalostí nezbytné. Kupříkladu bude nutné, aby databáze rodinných vztahů obsahovala generický koncept [RODIČE\_NĚJAKÉ\_OSOBY], jinak by báze musela obsahovat kompletní genealogii až k Adamovi a Evě. (Sowa, 1984; Nguyen a Corbett, 2006)

### Individuální koncepty

Individuální koncept má v poli referenta individuální označovač, který specifikuje unikátní koncept v katalogu individuů například ve znalostní bázi: např. [STUDENT\_UK: #35782557], kdy řetězec znaků ukazuje na právě jednoho studenta. Individuální označovač nahrazuje nějaké individuum z reálného vnímaného nebo hypotetického světa. Jakýkoliv typ konceptu může mít individuální označovač: například koncept [KOČKA] nebo [KOČKA: \*] referuje o nespécifikované kočce, ale [KOČKA: #246767] referuje o určitém individuu se sériovým číslem. Koncept [SKOK] referuje o nespécifikovaném aktu skákání, ale [SKOK : #5687] referuje o určité instanci skákání , která má nějaké vlastnosti (například výšku, trajektorii atd.) Individuální koncepty mohou referovat množná látková podstatná jména jako je voda nebo máslo nebo počítatelná podstatná jména jako jsou tužky nebo kluci. Koncept [VODA: #35755] bude reprezentovat určité množství vody. Pole referenta musí označovat individuum. Může to být například i [KOČKA:Micka], ale jen pokud je Micka unikátním identifikátorem v daném systému a je absolutně zřejmé o jakou kočku jménem Micka jde. (Mineau, 1996, str. 56; Sowa, 1984, str. 85)

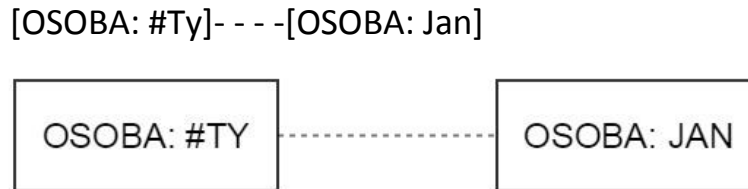
#### 2.3.6 Koreferenční spojení

V konceptuálním grafu mohou dva nebo více konceptů referovat o stejném individuu. Zavádí se tak pojem koreferenční spojení, jenž se v grafu značí přerušovanou čarou, a které říká, že pojmové obsahy propojených konceptů jsou totožné. (ISO/IEC 24707, 2007)

Koreferenční soubor **C** v konceptuálním grafu **G** je soubor konceptů vybraných z **G** nebo z grafů zahrnutých v kontextech grafu **G**. a platí pro něj:

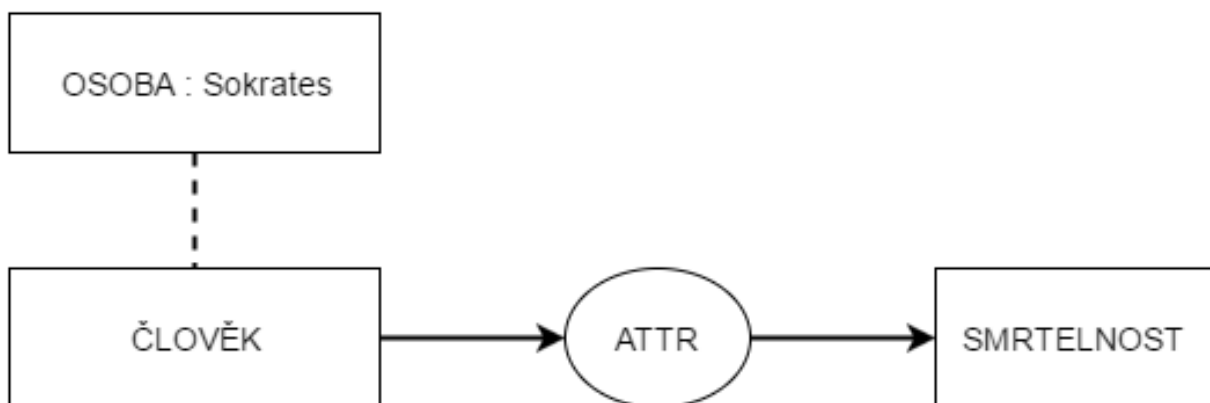
- Pro jakýkoliv koreferenční soubor **C** musí existovat alespoň jeden koncept v **C**, který se nazývá dominantní uzel **C** a který obsahuje všechny koncepty v **C**, jež pod něj spadají. Všechny dominantní uzly v **C** musejí být tzv. spoluzahrnuté (viz kapitola 2.5).
- Pokud je koncept **c** dominantním uzlem koreferenčního souboru **C** neměl by být prvkem jakéhokoliv jiného koreferenčního souboru.
- Koncept **c** může být členem více než 1 koreferenčního souboru **{C1, C2....}**, za předpokladu, že není **c** dominantním uzlem žádného **Ci**.

- Koreferenční soubor **C** se může skládat z jediného konceptu **c**, který se poté nazývá dominantním uzlem **C**. (Sowa, 1984, 2000)



Obr. 2.7 Lineární a grafické znázornění koreferenčního spojení 1 (Sowa, 1984)

**Obr. 2.7 Lineární a grafické znázornění koreferenčního spojení 1** zobrazuje koreferenční spojení v grafické i lineární formě mezi koncepty [OSOBA: #Ty] a [OSOBA : Jan] reprezentuje vyjádření: „Existuje osoba „Ty“ a existuje osoba „Jan“ . Tyto dva koncepty jsou koreferenční. Ty jsi Jan.“

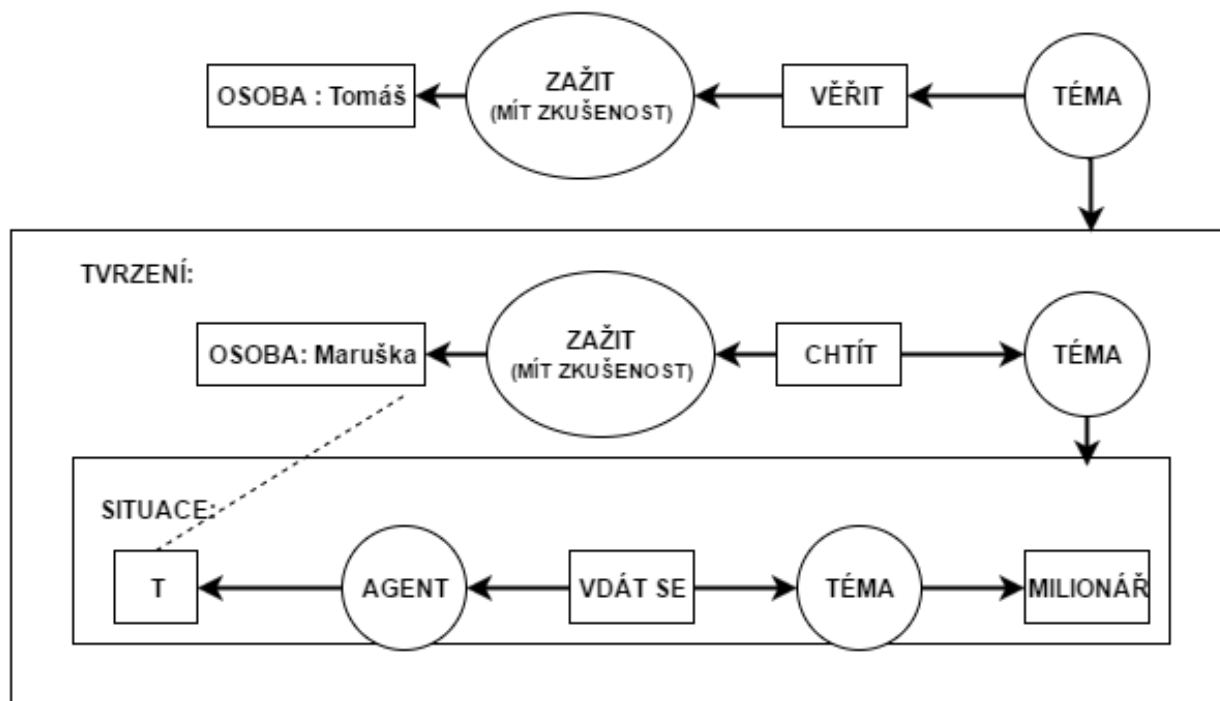


Obr. 2.8 Grafické znázornění koreferenčního spojení 2 (Sowa,1984)

Konceptuální graf na Obr. 2.8 reprezentuje tvrzení, že: Existuje člověk, jenž má atribut „Smrtnost“. Koncept [OSOBA : Sokrates] je spojen přerušovanou čarou s konceptem [ČLOVĚK]. Tyto koncepty tedy představují stejnou entitu. z toho můžeme vyvodit tvrzení: „Sokrates je smrtelný člověk.“ v lineární formě by tato věta byla zapsána následovně:

[OSOBA: Sokrates] – – – [ČLOVĚK] → (ATTR) → [SMRTELNOST]

Vzor. 2.4 Lineární znázornění koreferenčního spojení (Sowa,1984)



Obr. 2.9 Grafická forma konceptuální grafu se dvěma zahnížděnými grafy – koncepty a koreferenčním spojením reprezentující větu: „Tomáš věří, že se Maruška chce vdát za milionáře.“ (Sowa, 2000, str. 458)

Konceptuální graf se dvěma zahnížděnými grafy – koncepty a koreferenčním spojením reprezentující větu: „Tomáš věří, že se Maruška chce vdát za milionáře.“ na Obr. 2.9 a Vzor. 2.5 by se dal popsat následovně: „[Tomáš] má zkušenost s konceptem [VĚŘIT], který je propojen vztahem (TÉMA) s konceptem [TVRZENÍ], které je definováno pomocí zahnížděného konceptuálního grafu. Tento zahnížděný graf tvrdí, že [Maruška] má zkušenost s konceptem [Chtít], který je spojen vztahem (TÉMA) s konceptem [SITUACE]. Koncept [SITUACE] je popsán dalším zahnížděným konceptuálním grafem, který říká, že Maruška (reprezentována konceptem T) se vdá za [MILIONÁŘE].

Koreferenční linka spojuje koncepty [OSOBA: MARUŠKA] a [T]. Koncept [T] spadá do působnosti dominantního uzlu [OSOBA: MARUŠKA]. Jelikož oba dva koncepty reprezentují stejnou individualitu, měla by být jakákoliv informace z dominantního uzlu zkopírována do uzlu druhého. Návěští typu OSOBA nebo referent Marie mohou být například kopírovány z dominantního uzlu do konceptu [T].

[OSOBA : Tomáš] ← (ZAŽÍT) ← [VĚŘIT] → (TÉMA) –  
 [TVRZENÍ: [OSOBA: Maruška \* x] ← ( VĚŘIT) ← [CHTÍT] → (TÉMA) –  
 [SITUACE: [? x] ← (AGNT) ← [VDÁT SE] → (TÉMA) → [MILIONÁŘ] ] ]

*Vzor. 2.5 Lineární forma konceptuální grafu se dvěma zahnížděnými grafy – koncepty a koreferenčním spojením reprezentující větu: „Tomáš věří, že se Maruška chce vdát za milionáře.“ (Sowa, 2000, str. 458)*

V lineární notaci mohou být dva koncepty spojeny přerušovanou čarou jen v případě, že jsou zapsány vedle sebe na jednom řádku. Pokud tomu tak není (viz Vzor. 2.5), musí být koreferenční soubor označen koreferenčními návěštími, která se řídí stejnými konvencemi jako koreferenční návěští užívané v existenciálních grafech. Jeden z dominantních uzlů je označen definujícím návěštím s prefixem hvězdičky a další uzly jsou označeny vázanými návěštími s prefixem otazníku. v jakémkoliv libovolném koreferenčním souboru může být pouze jeden dominantní uzel označen s definujícím návěštím, avšak během dokazování nebo výpočtů je umožněno posunout definující návěští z jednoho dominantního uzlu na jiný.

### 2.3.7 Konceptuální vztahy

Konceptuální vztahy jsou v grafech značeny kruhem nebo oválem, v lineární notaci jsou zapisovány do kulaté závorky. Konceptuální vztah  $r$  je definován pomocí *typu vztahu*  $t$ , *valence*  $n$  a *signatury*  $s$ . Všechny konceptuální vztahy, které mají stejný typ vztahu  $t$ , mají stejnou valenci  $n$  a stejnou signaturu  $s$ . Typem vztahu je jméno vztahu. Každý konceptuální vztah má vždy typ vztahu, avšak nemusí mít referent. Typ vztahu také určuje valenci a signaturu, tj. počet hran k němu příslušných a typy konceptů, které se k němu vztahují. (Sowa, 2000, str. 478)

### 2.3.8 Valence vztahu

Valence vztahu je nezáporné celé číslo, které odpovídá počtu hran spojujících konceptuální vztah s koncepty. Pokud je tedy například valence  $n=2$ , vztah má dvě hrany a je propojen s dvěma koncepty. Vztah s valencí  $n$  se nazývá  $n$ -adický vztah a konceptuální vztah s valencí ( $n = 1$ ) monadický, dyadický ( $n = 2$ ), triadický ( $n = 3$ ). (Sowa, 2001)

### 2.3.9 Signatura vztahu

Signatura vztahu je seznam typů konceptů náležících danému vztahu. Signatura vztahu je sekvence  $\langle t_1, \dots, t_n \rangle$  typů konceptů připojených ke vztahu. 0-adický konceptuální vztah nemá žádné hrany a tudíž i jeho signatura je prázdná, protože k němu nejsou připojené

žádné koncepty. Počet záznamů v signatuře je číslo  $n$ , které má stejnou hodnotu jako valence vztahu. (Sowa, 2001)

### 2.3.10 Typ vztahu

Obecně je možno říci, že typ vztahu říká, o jaký druh vztahu se jedná. Typ vztahu určuje valenci vztahu a jeho signaturu. Typ vztahu také definuje počet připojených hran a určuje typy konceptů, které jsou k těmto hranám připojeny. Všechny konceptuální vztahy stejného typu vztahu  $t$  mají stejnou valenci  $n$  a stejnou signaturu  $s$ . Pro každý  $n$ -adický konceptuální vztah  $r$ , je typ vztahu buď návěštím vztahu z množiny  $R$  o valenci  $n$  nebo  $n$ -adickým lambda výrazem (viz kapitola 0). (Sowa, 2000, 2001)

Jak bylo již specifikováno v kapitole 2.3.2, typy vztahů jsou hierarchicky uspořádány do množiny  $R$ , což je částečně uspořádaná množina, jejíž prvky se nazývají *návěští typů vztahů*. Částečné uspořádání množiny návěští vztahů musí být konzistentní se závěry definovanými lambda výrazy. Každé návěští typu vztahů je specifikováno jako primitivní nebo jako definované. (Sowa, 2000, 2001)

#### Druhy návěští:

- **Primitivní návěští**

Primitivní návěští typu vztahu jsou pouhými jmény zařaditelnými do hierarchie  $R$ . Jmen vztahů je mnoho. v této práci jsem užívala například (AGENT), (ATRIBUT) nebo (MÍSTO) Celý seznam navrhl Sowa ve své knize Conceptual Structures (1984, str. 415 - 419), ale je samozřejmě možné je nadefinovat a pojmenovat libovolně.

- **Definované návěští určené lambda výrazem**

Definovaná návěští jsou určována pomocí  $n$ -adických lambda (viz kapitola 2.4.2) výrazů a sestávají ze jména a  $n$ -adického lambda výrazu. Pro každé definované návěští vztahu o valenci  $n$ , existuje právě jeden  $n$ -adický lambda výraz nazvaný „jeho definice“. Definované návěští vztahu a jeho definice jsou zaměnitelné. Na jakékoliv pozici v konceptuálním grafu, kde se vyskytuje jeden, může být nahrazen druhým. Například typ vztahu „JETNĚKAM“ (GOING TO) může být definován konceptuálním grafem nebo můžeme říci rovnicí. Díky tomu je typ vztahu „JETNĚKAM“ synonymem pro dyadický lambda výraz. (Sowa, 1999)



1. [VZTAH: JETNĚKAM] → (DEF) → [LAMBDAVÝRAZ:  
[OSOBA: λ1] ← (AGNT) ← [JET] → (MÍSTO) → [MĚSTO: λ2] ].
2. JETNĚKAM = [OSOBA: λ1] ← (AGNT) ← [JET] → (MÍSTO) → [MĚSTO: λ2].

*Vzor. 2.6 Dva rozdílné zápisy definovaného návěští určeného pomocí n-adických lambda výrazů (Sowa, 1999)*

Definice zobrazená ve Vzor. 2.6 říká, že vztah (JETNĚKAM) je definován lambda výrazem, který vztahuje osobu označenou jako  $\lambda_1$ , jenž je agentem (AGNT) konceptu [JET] s městem označeným jako  $\lambda_2$ , což je (MÍSTO) spojené s [JET]. Pomocí tohoto abstraktního vztahu je možno vytvořit reprezentaci věty „Jan jede do Brna“ v podobě konceptuálního grafu ve formuli Vzor. 2.7.

[OSOBA: Jan] → (JETNĚKAM) → [MĚSTO: Brno].

*Vzor. 2.7 Lineární zápis věty „Jan jede do Brna“ (Sowa, 1999)*

Graf ve formuli Vzor. 2.7. může být rozšířen na detailnější graf tím, že se návěští typu vztahu (JETNĚKAM) nahradí jeho definicí jak je vidět ve Vzor. 2.8.

[OSOBA: Jan] → ([OSOBA: λ1] ← (AGNT) ← [JET] –  
→ (MÍSTO) → [MĚSTO: λ2]) → [MĚSTO: Brno].

*Vzor. 2.8 Rozšířený lineární zápis věty „Jan jede do Brna“ (Sowa, 1999)*

Dalším krokem je odstranit lambda výraz z kruhu nebo z kulatých závorek, a sjednotit první parametr [OSOBA: λ1] s konceptem připojeným k první hraně a sjednotit druhý parametr [MĚSTO: λ2] s konceptem připojeným k druhé hraně. Výsledkem je graf ve formuli Vzor. 2.9. Tento graf říká: „Osoba Jan je agentem JET a Město Brno je cílem Jetí.“ Každý krok této derivace může být v tomto finálním rozšířeném grafu proveden opačně k derivaci grafu původního.

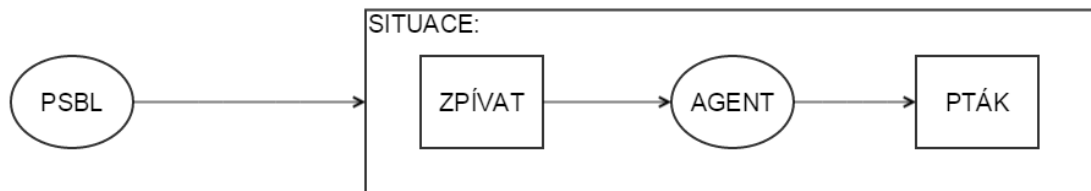
[OSOBA: Jan] ← (AGNT) ← [JET] → (MÍSTO) → [MĚSTO: Brno].

*Vzor. 2.9 Lineární zápis konceptuálního grafu vyjadřujícího: „Osoba Jan je agentem JET a Město Brno je cílem Jetí.“ (Sowa, 1999)*

### 2.3.11 Příklady konceptuálních vztahů s rozdílnou valencí

#### Monadický vztah

Monadický vztah je vztah o valenci 1, což znamená, že se k němu váže pouze jeden koncept. v tomto případě je to koncept s typem konceptu Situace a referentem, jež má definovaný designátor pomocí zahrnutého grafu. V příkladech na Vzor. 2.10 a na Obr. 2.10 níže je znázorněn konceptuální vztah vyjadřující možnost, podobným příkladem byl konceptuální vztah určující minulost (PAST).



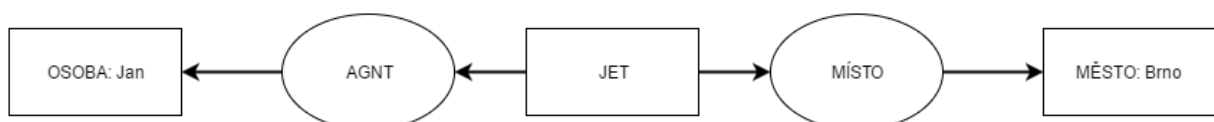
Obr. 2.10 Grafická forma grafu s monadickým vztahem pro větu: "Je možné, že pták zpívá."

(PSBL)→[SITUACE: [PTÁK]←(AGNT)←[ZPÍVAT] ]

Vzor. 2.10 Lineární forma grafu s monadickým vztahem pro větu: "Je možné, že pták zpívá."

#### Dyadický vztah

Dyadický typ vztahu má valenci 2 a váží se k němu tedy 2 koncepty.



Obr. 2.11 Grafická forma grafu s dyadickým vztahem pro větu „Jan jede do Brna“(Sowa, 1984)

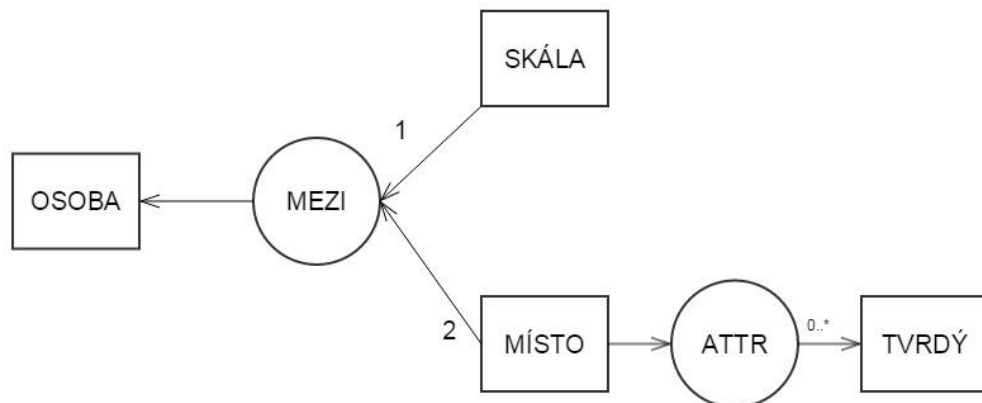
[OSOBA: Jan] ← (AGNT) ← [JET] → (MÍSTO) → [MĚSTO: Brno].

Vzor. 2.11 Lineární forma grafu s dyadickým vztahem pro větu „Jan jede do Brna" (Sowa, 1984)

Grafy na Obr.2.11 a Vzor.2.1 reprezentují: [OSOBA: Jan] je Agentem konceptu [JET] a [MĚSTO : Brno] je Místem konceptu [JET]. v tomto grafu jsou (AGENT) a (MÍSTO) dyadickými typy vztahu. Signatura vztahů reprezentuje omezení na typy konceptů, které mohou být spojeny s jejich hranami. Kupříkladu signatura typu vztahu (AGENT) obsahuje prvky <Jednat, Živoucí>, což indikuje, že typ konceptu spojený s první hranou musí být typ konceptu JEDNAT nebo podtyp jako je například JET a typ konceptu spojený s druhou hranou musí být ŽIVOUČÍ nebo nějaký podtyp, jako je v tomto případě OSOBA. (Sowa, 1984)

### Triadický vztah

Triadický vztah má valenci i signaturu. k danému vztahu se tedy váží 3 koncepty.



Obr. 2.12 Grafická forma grafu s triadickým vztahem pro větu „Osoba je mezi skálou a tvrdým místem.“ (Sowa, 1984)

[OSOBA] ← (MEZI) –

← 1 – [SKÁLA]

← 2 – [MÍSTO] → (ATTR) → [TVRDÝ].

Vzor. 2.12 Lineární forma grafu s triadickým vztahem pro větu „Osoba je mezi skálou a tvrdým místem.“ (Sowa, 1984)

Graf na Obr. 2.12 Vzor. 2.12 reprezentují: Pro konceptuální vztah (MEZI) o valenci 3 je signatura <Entita, Entita, Entita>, která říká, že všechny tři připojené koncepty musejí být typu entita, tedy nejobecnějším typem, který neukládá žádná omezení. První dvě hrany

ukazují ke dvěma konceptům typu entita (SKÁLA a MÍSTO) a nacházejí se na opačné straně nežli třetí koncept typu entita (OSOBA). (Sowa, 1984)

### 2.3.12 Konvence směru hran v konceptuálních grafech

Pro konceptuální vztahy, které mají  $n$  hran, platí, že prvních  $n-1$  hran má šipky ukazující směrem ke konceptuálnímu vztahu značeného kruhem nebo závorkou a  $n$ -tá či můžeme říci poslední hrana směřuje směrem od vztahu v kruhu.

Pomlčka v lineárním zápisu za konceptuálním vztahem (například za vztahem (MEZI) ve Vzor. 2.12) znamená, že na následujících řádcích se nacházejí další hrany. Dvě hrany směřující směrem ke vztahu jsou číslovány 1 a 2. Pro monadické vztahy je zápis čísla 1 a šipka směřující ke vztahu nepovinná. Pro dyadické vztahy jsou hrany buď číslovány 1 a 2 nebo první hrana míří směrem ke vztahu a druhá směřuje od něj. (Sowa, 1984)

## 2.4 Lambda kalkulus a lambda výrazy v konceptuálních grafech

Krom základních operátorů a kvantifikátorů vyžaduje logika často i metodu definice nových vztahů. John F. Sowa ve svých konceptuálních grafech využívá tzv. lambda kalkulu. Lambda kalkulus je systematická metoda pro vývoj a deklaraci funkcí a vztahů s pravidly pro jejich vyhodnocování. Jeho první verze pochází ze třicátých let 20. století. Popsal jej Alonzo Church roku 1941, když se snažil vytvořit nový funkcionální základ pro matematiku. k tomuto účelu sice jeho systém použit zatím nebyl, ale našel využití jinde. Jde o výrazy, které umožňují formulovat obecná tvrzení. v teorii konceptuálních grafů se užívá lambda výrazů k definici nových typů konceptů a nových typů konceptuálních vztahů. (Sowa, 2000, str. 154-155)

### 2.4.1 Definice lambda výrazu

Lambda výraz je konceptuální graf s 0 či více koncepty, které mají své referenty nahrazeny znakem lambda  $\lambda$ . Pro každé přirozené číslo  $n$ , je  $n$ -adický lambda výraz  $e$  konceptuálním grafem nazvaným „tělem  $e$ “, ve kterém bylo  $n$  konceptů označeno jako formální parametry  $e$ . Tyto formální parametry jsou číslovány od 1 do  $n$ . Existuje sekvence  $\langle t_1, \dots, t_n \rangle$ , nazývána signatura  $e$ , kde je každé  $t_i$  je typem konceptu  $i$ -tého formálního parametru  $e$ . Signatura je tedy jednoduše jen seznam typů formálních parametrů. Speciálním případem je 0-adický lambda výraz. Jelikož 0-adické lambda výrazy nemají žádné formální

parametry, jejich signatura je prázdná sekvence značená  $\langle \rangle$ . Není žádný sémantický rozdíl mezi konceptuálním grafem bez formálních parametrů a 0-adickým lambda výrazem. Lambda výrazy s jedním formálním parametrem se nazývají monadické a samozřejmě existují i dyadické, triadické až n-adické. (Sowa, 2001)

### **Formální parametr**

Jak bylo řečeno, lambda výrazy jsou konceptuální grafy s 0 či více koncepty označenými jako *formální parametry*. Formální parametr je vlastně syntaktickým konstruktem, jenž je zástupným symbolem pro referent jiného konceptu. Nebo je také možno říci, že je formální parametr pojem v lambda výrazu, jehož referent není definován, dokud není koncept spojen s nějakým koreferenčním souborem, jehož referent definován je. Formální parametr je parametr použitý při psaní funkce; její vnitřní proměnná. Ta je vždy před zpracováním nahrazována hodnotou *skutečného parametru*. Skutečný parametr je proměnná nebo výraz dosazený při volání funkce. Při volání funkce je jeho hodnota přiřazena formálnímu parametru. (Petersen, 2005)

Tato abstraktní definice nspecifikuje, jak jsou formální parametry označeny, není tedy nijak specifikováno, jak formální parametry označovat. Tradiční notace, kterou představil logik Alonzo Church (1941), používá pro označení parametrů řecká písmena lambda  $\lambda$ . Pokud je  $n > 1$  mohou být formální parametry označeny množinou  $\{\lambda_1, \lambda_2, \dots, \lambda_n\}$ . Často se pro zjednodušení zápisu (například v rámci CGIF notace) upouští od užití písmene lambda a spíše se užívá  $?x, ?y, ?z, ?a, ?b$  (Sowa, 1999).

### **Obecná struktura lambda výrazu:**

#### **Jmenná část(\*x) = část definice(konceptuální graf )**

Formální parametr v levé jmenné části, který se označuje **(\*x\*y\*z)** se nazývá *definující návěští* a formální parametr **?x** v pravé části rovnice se nazývá *vázané návěští*. Definované návěští říká, s jakými proměnnými se bude pracovat v definující části.

### Příklad užití lambda kalkulu:

Převedení věty „Jan jede do Brna“ do dyadického lambda výrazu, který definuje konceptuální vztah spojující osobu a město (JETNĚKAM), je dosaženo tím, že je zaměněno jméno Jan za symbol  $?x$  a jméno Brno za  $?y$ . Jelikož se jedná o dyadický lambda výraz, jeho signatura je seznam dvou typů  $\langle \text{Osoba}, \text{Město} \rangle$ , který pochází z typů polí formálních parametrů.

#### **1. Definice vztahu (JETNĚKAM)**

Vztah JETNĚKAM ( $*x, *y$ )

$[\text{OSOBA}: ?x] \leftarrow (\text{AGNT}) \leftarrow [\text{JET}] \rightarrow (\text{MÍSTO}) \rightarrow [\text{MĚSTO}: ?y]$

#### **2. Definice vztahu (JETDOBRNA)**

Vztah JetDoBrna( $*x$ )

$[\text{OSOBA}: ?x] \leftarrow (\text{AGNT}) \leftarrow [\text{JET}] \rightarrow (\text{MÍSTO}) \rightarrow [\text{MĚSTO}: \text{Brno}]$

#### **3. Definice vztahu (JETDOBRNA) pomocí prvního lambda výrazu z prvního bodu**

Vztah JetDoBrna( $*x$ )

$[\text{OSOBA}: ?x] \rightarrow (\text{JETNĚKAM}) \rightarrow [\text{MĚSTO}: \text{Brno}]$

Vzor. 2.13 Využití lambda kalkulu při definici konceptuálních vztahů (JETNĚKAM) a (JETDOBRNA) (Petersen, 2005)

$[\text{PROFESOR} : \text{Červíček}] \leftarrow (\text{JETDOBRNA})$

Vzor. 2.14 Lineární forma grafu využívající vztah definovaný lambda kalkulem (JETDOBRNA) reprezentující větu „Profesor Červíček jede do Brna“. (Petersen, 2005)

V konceptuálním grafu zapsaném Vzor. 2.14 reprezentující větu „Profesor Červíček jede do Brna“ se koncept  $[\text{PROFESOR} : \text{Červíček}]$  stává tzv. *aktuálním parametrem* lambda výrazu z formule Vzor. 2.13. Typ konceptu PROFESOR je také podtypem typu konceptu OSOBA, což je typ konceptu původní definice ve formuli Vzor. 2.13., který je nahrazen aktuálním parametrem  $[\text{PROFESOR} : \text{Červíček}]$ . Typ aktuálního parametru musí být tedy podtypem formálního parametru.

## 2.4.2 Lambda výraz pro definici typů konceptu

V případě nutnosti definovat nové návěští typu konceptu, je užíváno monodických lambda výrazů (Petersen, 2005). Takovou definici ilustruje následující příklad

### 1. Definice typu konceptu [TRAGICKYMILENEC]

Typ konceptu TragickyMilenec(\*x)

[MILENEC: ?x]→(ATTR)→[TRAGICKY]

### 2. Využití typu konceptu [TRAGICKYMILENEC] pro reprezentaci věty: „Dvojice tragických milenců Romeo a Julie si bere svůj život.“

[Tragickymilenec: {Romeo,Julie}@2<sup>13</sup>]←(AGNT)←[VZÍT]-

→(THME)→[ŽIVOT: #jejich<sup>14</sup>]

Vzor. 2.15 Využití lambda kalkulu pro definici typu konceptu. Příklad je autorskou adaptací příkladu z (Petersen, 2005)

---

<sup>13</sup> @2 říká, že v seznamu referentů jsou dvě entity

<sup>14</sup> #jejich: znak # znamená že se jedná o zájmeno

### 2.4.3 Lambda výraz pro definici vztahů

Lambda kalkul se užívá pro definici definované návěští typu vztahu, které se sestávají ze jména a n-adického lambda výrazu (Petersen, 2005) Takovou definici ilustruje následující příklad.

#### 1. Definice typu vztahu (ZPÍVÁNÍ)

V tomto vyjádření grafu představuje symbol  $*x$  formální parametr vztahu. Jak bylo vysvětleno formální parametr je zástupný symbol pro to, co nahradí symbol  $?x$  ve chvíli, kde bude vztah využit.

Vztah Zpívání( $*x$ )

[ŽIVOUCÍ:  $?x$ ] ← (AGNT) ← [ZPÍVAT]

#### 2. Využití typu vztahu (ZPÍVÁNÍ) pro reprezentaci věty: „Jan zpívá“

[OSOBA: Jan] ← (ZPÍVAT)

#### 3. Rozšíření vztahu (ZPÍVÁNÍ) o jeho lambda výraz ve větě: „Jan zpívá“

[OSOBA: Jan] je formálním parametrem lambda výrazu „ŽIVOUCÍ ENTITA ZPÍVÁ“ a OSOBA je podtypem typu ŽIVOUCÍ.

[OSOBA: Jan] ← ([ŽIVOUCÍ:  $?x$ ] ← (AGNT) ← [ZPÍVAT])

#### 4. Reprezentace věty „Jan zpívá“ za pomoci konceptu [Zpívat]

[OSOBA: Jan] ← (AGNT) ← [ZPÍVAT]

Vzor. 2.16 Využití lambda kalkulu pro definici typu konceptuálního vztahu. Příklad je autorskou adaptací příkladu z (Petersen, 2005)

## 2.5 Zahnížděné grafy a koncepty

Mohou existovat koncepty a konceptuální grafy, které jsou zahnížděné v jiných konceptech, což by bylo možno vysvětlit tak, že jsou vloženy v jiných konceptech.

### 2.5.1 Zahnížděné grafy

Pokud je konceptuální graf **G** v poli referentu konceptu **C**, tak je možné říci, že je graf **G** zahnížděný v konceptu **C**. Je možné vrstvit a vkládat konceptuální grafy do jakékoliv hloubky úrovní grafu: Graf **G** může být zahnížděný v konceptu z kontextu **C**, který může být



zahnížděný v nějakém konceptu třetího kontextu  $C_2$ . Graf  $G$  se tedy nazývá zahnížděný v konceptu  $C$  ve dvou případech:

- $G$  je přímo součástí referentu  $C$
- $G$  je přímo částí referentu konceptu  $C_2$ , který je uhnížděn uvnitř konceptu  $C$

Graf  $G$  je nazýván „bezprostředně zahnížděnými“ v  $C$ , když je jakýkoliv koncept  $c$  konceptuálního grafu  $G$  také „bezprostředně zahnížděný“ v  $C$ . Takovýto bezprostředně zahnížděný graf je referentem typu deskriptor. (Sowa 1984, 2000, 2001)

### 2.5.2 Zahnížděné a spolu – zahnížděné koncepty

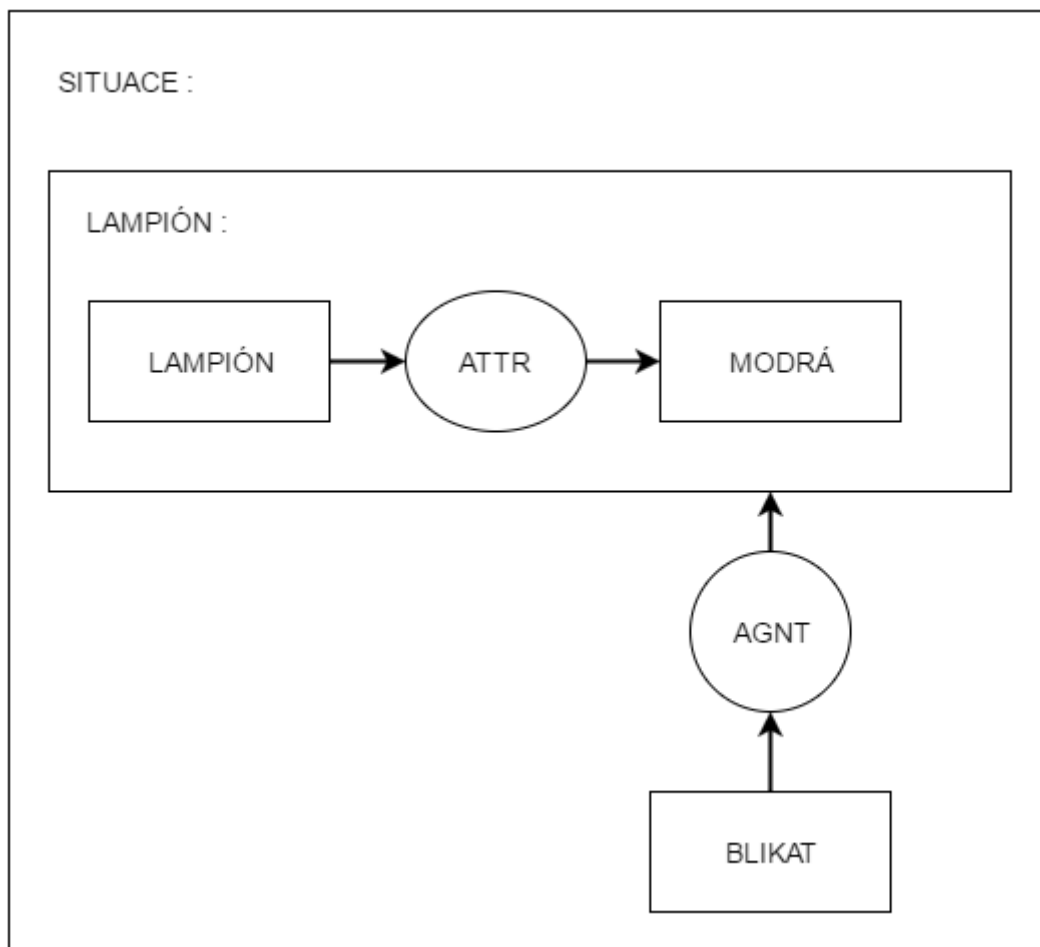
Koncept  $c$  se nazývá zahnížděným v  $C$ , pokud je  $c$  bezprostředně zahnížděné v  $C$  nebo je  $c$  bezprostředně zahnížděné v nějakém kontextu  $D$ , který je zahnížděný v  $C$ . (Sowa, 2001; Petersen, 2005)

- Dva koncepty  $c$  a  $d$  se nazývají spolu-zahnížděné, pokud se  $c=d$  nebo existuje nějaký kontext  $C$ , ve kterém jsou  $c$  a  $d$  bezprostředně zahnížděné. Jsou tedy buď totožné, nebo se nacházejí na stejné úrovni zahníždění.
- Pokud je koncept  $x$  spolu-zahnížděný s kontextem  $C$ , tak jakýkoliv koncept  $d$  zahnížděný v  $C$  se nazývá hlouběji zahnížděný nežli  $x$ .
- Koncept  $d$  se nazývá „uvnitř rámce“ konceptu  $c$ , pokud je koncept  $d$  spolu-zahnížděný s  $c$  nebo je  $d$  hlouběji zahnížděn nežli  $c$ . (Sowa, 2001)

### 2.5.3 Příklad zahnížděného konceptuálního grafu (Petersen, 2005):

[SITUACE: [LAMPIÓN: [LAMPIÓN] → (ATTR) → [MODRÁ]] –  
← (AGNT) ← [BLIKAT] ]

Vzor. 2.17 Příklad konceptuálního grafu s užitím zahnížděného grafu reprezentující větu „Modrý lampión bliká.“ (Petersen, 2005).



Obr. 2.13 Příklad konceptuálního grafu s užitím zahnížděného grafu vytvořený na základě příkladu lineárního zápisu věty „Modrý lampión bliká.“ (Petersen, 2005)

**Rozbor grafu z Vzor. 2.17 a Obr. 2.13 :**

V tomto konceptuálním grafu jsou dvě úrovně zahníždění:

[LAMPÍÓN]→(ATTR)→[MODRÁ]

je graf je zahnížděný v konceptu (kontextu):

[LAMPÍÓN: [LAMPÍÓN]→(ATTR)→[MODRÁ] ],

který je částí většího grafu

[LAMPÍÓN: [LAMPÍÓN]→(ATTR)→[MODRÁ] ]-

←(AGNT)←[BLIKAT] ],

který je následně zahnížděn v konceptu (kontextu) [Situace].

Koncept [BLIKAT] a vnější koncept [LAMPÍÓN] jsou spolu-zahnížděné. Vnitřní [LAMPÍÓN] a koncept [MODRÁ] jsou také spolu-zahnížděné koncepty. Avšak koncept

[MODRÁ] není spolu-zahnížděný s [BLIKAT], protože tyto koncepty nejsou bezprostředně zahrnuté ve stejném konceptu. (Petersen, 2005)

#### 2.5.4 Kontext

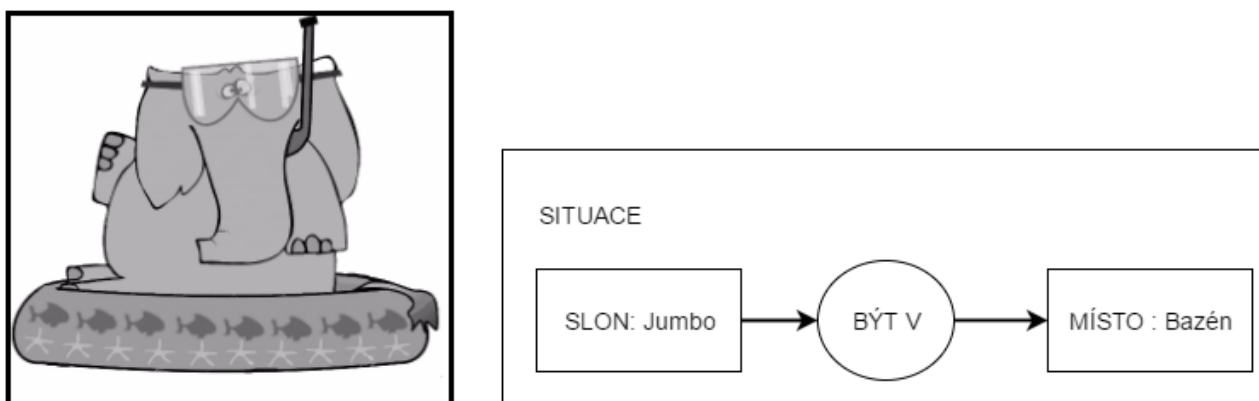
Kontext **C** je koncept, jehož designátorem (což je část referentu) je neprázdný konceptuální graf **G**. Kontext je tedy koncept se zahrnutým konceptuálním grafem, který popisuje referent (druh designátoru nazývaný deskriptor). Typickými příklady kontextů, ve kterých se objevují jiné grafy (podgrafy) jsou koncepty typu [TVRZENÍ] a [SITUACE] (Petersen, 2005). Můžeme také říci, že u kontextů se jedná o konjunkci grafů (Alhajj, Rokne, 2014) Všechny grafy zapsané ve vzorci ukazují typ konceptu TVRZENÍ a v poli referentu je zahrnutý graf, který specifikuje, co je obsahem tohoto tvrzení. Graf na obr. zobrazuje grafickou formu grafu, který obsahuje typ konceptu SITUACE a v poli referenta zahrnutý graf popisující situaci, ve které Jan jede do Brna. (Sowa, 1984)

[TVRZENÍ: [SLON: √] → (ATTR) → [BARVA : Šedá ]

Vzor. 2.18 Konceptuální graf s kontextem pro koncept typu [TVRZENÍ] pro větu „Všichni sloni jsou šedí.“ (Petersen, 2005)

[TVRZENÍ: [SLON: Jumbo] → (ATTR) → [BARVA: Šedá ]

Vzor. 2.19 Konceptuální graf s kontextem pro koncept typu [TVRZENÍ] pro větu „Jumbo je šedý slon.“ (Petersen, 2005)



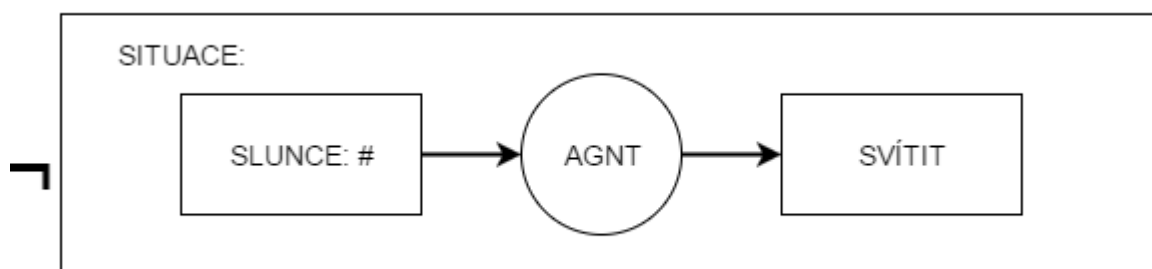
Obr. 2.14 Obrázek nalevo je reprezentován konceptuálním grafem s kontextem pro koncept typu [SITUACE]. Situace je popsána větou: „Slon Jumbo je v bazénu.“

## 2.6 Základní logické operátory v konceptuálních grafech

Každý konceptuální graf lze převést do formule predikátové logiky. (Lukasová, 2010)

### 2.6.1 Negace v konceptuálním grafu

v konceptuálním grafu se symbol negace umísťuje výhradně před kontexty (Situace, Tvzení....)



Obr. 2.15 Grafická forma konceptuálního grafu s negací pro větu „Slunce nesvítil“ odvozeno z lineární formy. (Lukasová, 2010)

$\neg$  [TVRZENÍ: [MUŽ] ← (EXPR) ← [ROZUMĚT] → (THME) → [ŽENA : {\*}]]

Vzor. 2.20 Lineární forma konceptuálního grafu s negací pro větu „Není pravda, že existuje muž, který rozumí ženám.“ (Lukasová, 2010).

### 2.6.2 Konjunkce v konceptuálním grafu

V konceptuálním grafu se konjunkce vyjadřuje výhradně v rámci téhož kontextu, a to bez spojovacích vztahů nebo linií (Lukasová, 2010).

[TVRZENÍ: [ŽENA : \* x] → (ATTR) → [KRÁSNÁ]

[ŽENA : \* x] → (ATTR) → [NEBEZPEČNÁ]

Vzor. 2.21 Lineární forma konceptuálního grafu s konjunkcí pro větu „Existuje žena, která je krásná a nebezpečná.“ (Lukasová, 2010)

### 2.6.3 Disjunkce v konceptuálním grafu

V konceptuálním grafu se disjunkce **P v G** řeší pomocí konjunkce a negace tzv. DeMorganovým pravidlem, které tvrdí že: **P v G** je ekvivalentní  $\neg(\neg P \wedge \neg G)$

#### Postup :

1. Negace kontextu okolo **P**
2. Negace kontextu okolo **G**
3. Zařazení **P a G** do třetího společného kontextu
4. Negace třetího společného kontextu

$\neg$ [SITUACE:

$\neg$ [SITUACE: [OSOBA: Jan] – – – [BLÁZEN]]

$\neg$ [SITUACE: [OSOBA: Jan]  $\rightarrow$  (ATTR)  $\rightarrow$  [CHYTROST]  $\rightarrow$  (MÍRA)

$\rightarrow$  [ STUPEŇ: #VELMI]]

]

Vzor. 2.22 Příklad řešení disjunkce DeMorganovým pravidlem: „Jan je buď velmi chytrý, nebo blázen.“ (Lukasová, 2010)

### 2.6.4 Implikace konceptuálních grafů

Implikace v konceptuálních grafech lze zobrazit jako kombinaci negace a konjunkce (Kent, Williams, 1996). Jinak řečeno implikace vznikne přidáním negace před konceptuální graf, který obsahuje kontext a následně i před konceptuální graf, který znázorňuje kontext. Konceptuální graf na Vzor. 2.23 doslovně reprezentuje tvrzení, že „Není pravdou, že existuje člověk x, který nemá atribut smrtelnosti.“ Což může být vyjádřeno implikací: Pokud existuje člověk **x**, tak je **x** smrtelný.

$\neg$  [ [ČLOVĚK: \* x]

$\neg$  [ [? x]  $\rightarrow$  (ATRIBUT)  $\rightarrow$  [SMRTELNOST] ]].

Vzor. 2.23 Lineární forma konceptuálního grafu s implikací (Sowa, 1984)

### 3 Vybraná rozšíření teorie konceptuálních grafů

Předchozí kapitola představila základní teorii konceptuálních grafů jako formalismu jazyka reprezentace znalostí. Užití konceptuálních grafů bylo od jejich představení velmi široké a s jejich využitím v praxi leckdy přineslo úplně nové principy či rozšíření. Tato kapitola tedy předkládá vybraná řešení, která vnášejí do teorie konceptuálních grafů buď zcela nové prvky (fuzzy logika) nebo nabízí elegantnější řešení (reprezentace času v konceptuálních grafech) nebo přispívá ke kvalitnější reprezentaci znalostí, nebo přináší formalizaci operací, popřípadě standardizovanou formu notace (CGIF). Některá rozšíření připojil sám John. F. Sowa a následně je rozpracovali jiní autoři a jiné jsou příspěvky jiných autorů. Rozšíření jako jsou funkce nebo fuzzy logika a samozřejmě pravidla pro tvorbu grafů a odvozování jsou přijímána komunitou užívající konceptuálních grafů a jsou často využívána v implementační rovině konceptuálních grafů.

#### 3.1 CGIF Conceptual Graph Interchange Format

Toto rozšíření spočívá ve standardizované notaci konceptuálních grafů, jež byla vyvinuta pro komunikaci mezi stroji a systémy. Tento speciální formát zápisu konceptuálních grafů má jednodušší syntax a omezenější znakovou sadu: CGIF (*Conceptual Graph Interchange Format*). CGIF byl standardizován normou ISO/IEC 24707:2007 pod ISO jako jeden ze tří dialektů (do kterých se kromě CGIF řadí CLIF a XCL) tzv. „Obecné logiky“ (Common Logic), která tvoří rámec skupiny jazyků založených na logice. Specifikace tohoto standardu zaručují, že jakákoliv věta vyjádřená v jednom z těchto tří jazyků může být přeložena do jiného v logicky ekvivalentní formě. Avšak i přesto, že překlad zachová sémantiku, nezaručuje to, že zachová veškeré syntaktické detaily: například věta přeložená z jednoho dialektu do druhého a zpět do původního bude logikou odpovídat originálu, ale některé podvýrazy mohou být v jiném pořadí nebo mohou být nahrazeny sémantickými ekvivalenty. (Sowa, 2008, str. 2018)

CGIF bylo vytvořeno jako formát pro výměnu mezi IT systémy, které používají konceptuální grafy jako svoji interní reprezentaci. Pro komunikaci se systémy, které používají jinou interní reprezentaci, může být CGIF přeložen do tzv. Knowledge Interchange Format (viz Vzor. 3.4). (Sowa, 2001)

### Existují 2 typy CGIF:

- **Core CGIF**

Beztypová verze logiky, která komplexně vyjadřuje sémantiku obecné logiky. Tento dialekt koresponduje s Piercovými existenciálními grafy: jsou to pouze primitiva a konjunkce, negace a existenciální kvantifikátor. Core CGIF je standardizovanou prostřednictvím ISO/IEC 24707

- **Rozšířené CGIF**

Core CGIF je rozšířen o univerzální kvantifikátor, booleovský kontext s návěštmi typu (If, Then, Either, Or, Equivalence, Iff) a možnost do CGIF importovat externí text. i přesto, že je rozšířené CGIF typovaný jazyk, není striktně typovaný, protože typová návěští se užívají jen pro omezený rozsah kvantifikátorů. Tento fakt zamezuje vzniku syntaxových chyb jako je u striktně typované logiky Core CGIF.

V CGIF jsou koncepty reprezentovány hranatými závorkami a konceptuální vztahy jsou reprezentovány závorkami kulatými. Řetězec znaků, který má jako předponu hvězdičku (například *\*x*), se nazývá *definující návěští*, na které může odkazovat tzv. *návazné návěští*, které má jako předponu otazník (*?x*). Tyto řetězce, které se v CGIF nazývají koreferenční návěští, odpovídají proměnným užívaným v jazyce CLIF (Common Logic Interchange Format). Pokud nemá definující návěští předponu *@every*, je definující návěští překládáno jako existenciální kvantifikátor.

Jak ukazují příklady CGIF a CLIF níže (viz *Vzor. 3.1*, *Vzor. 3.2*, ***Vzor. 3.3***), rozdíly mezi nimi vycházejí ze struktury grafu: uzly grafu nemají implicitně určené pořadí a koreferenční návěští jako *\*x* či *?x* představují spíše spojení mezi jednotlivými uzly, než-li proměnné. CGIF používá předpony *\** a *?* pro rozlišení koreferenčních návěští a konstant, kdežto CLIF žádnou podobnou syntaktickou konvenci neužívá. CGIF na rozdíl od CLIF neužívá operátor **AND**, protože konjunkce uzlů v rámci jakéhokoliv kontextu je implicitní. Vynechání operátoru konjunkce v CGIF často vede ke snížení počtu závorek.

### Příklady:

[JET: \* x] [OSOBA: Jan] [MĚSTO: Brno] [AUTOBUS:  
\* y] (AGNT ? x Jan) (MÍSTO ? x Brno) (NÁSTROJ ? x ? y)

*Vzor. 3.1 Lineární forma zápisu věty v CGIF: „Jan jede do Brna.“ (Sowa, 2001)*

[JET \* x] (AGNT ? x [Osoba: Jan]) (MÍSTO ? x [MĚSTO  
: Brno]) (NÁSTROJ ? x [AUTOBUS])

*Vzor. 3.2 Lineární forma formule 1.1 zkrácená pomocí zahnížděných grafů (Sowa, 2001)*

existuje ((x JET) (y AUTOBUS))

(and (OSOBA Jan) (Město Brno))

(AGNT x Jan) (MÍSTO x Brno) (NÁSTROJ x y)))

*Vzor. 3.3 Lineární forma zápisu věty v CLIF: „Jan jede do Brna.“ (Sowa, 2001)*

(existuje ((? x JET) (? y OSOBA) (? z MĚSTO) (? w AUTOBUS))

(AND (JMÉNO ? y Jan) (JMÉNO ? z Brno))

(AGNT ? x ? y) (MÍSTO ? x ? z) (NÁSTROJ ? x ? w)))

*Vzor. 3.4 Pro srovnání lineární forma zápisu věty v KIF: „Jan jede do Brna.“ (Sowa, 2001)*

## 3.2 Funkce

Funkce jsou speciálním typem vztahu konceptuálních grafů. v původní teorii, tak jak ji navrhl Sowa (1994, 2000) notace funkcí chyběly. Na absenci notace funkcí ve formalismu konceptuálních grafů upozornil v roce 1994 Guy W. Mineau ve své práci Views, Mappings and Functions: Essential Definitions to The Conceptual Graph Theory. Mineau (1994) tvrdí, že jedině konceptuální grafy schopné vyjadřovat funkční závislosti je možné použít pro modelování<sup>15</sup> znalostí. Pod takovýmto modelováním je možné představit si cokoliv; kupříkladu

---

<sup>15</sup> Podobor znalostního inženýrství, který se zabývá tvorbou znalostních modelů spíše než finální implementací znalostních systémů (Berka, 2007)



modelování je potřebné při návrhu softwaru, aby byl vznikající software kompatibilní s ostatním software, ale i hardwarem. (Cox, Delugach a Skipper 2001)

V neposlední řadě jsou funkce součástí mnoha jazyků reprezentace znalostí (např. Ontolingua) a pro validní mapování z jednoho jazyka do druhého, které je nezbytné pro výměnu znalostí mezi systémy užívající rozdílné jazyky reprezentace znalostí, je implementace funkce nezbytná.

### 3.2.1 Definice:

Funkce jsou speciálním typem vztahů (relací), kdy je poslední parametr funkčně závislý na jeho  $n-1$  předchozích parametrech. Funkční závislost konstatuje, že pokud  $x_1$  ( $x_2$ ) jsou propojené s  $y_1$  ( $y_2$ ) skrze funkci  $f$  a pokud je  $y_1$  funkčně závislé na  $y_1$  a  $x_1 = x_2$ , tak  $y_1 = y_2$ . (Kent a Williams, 1996, str. 57) Neformálně řečeno je funkce speciálním vztahem, který ukládá omezení, které spočívá v tom, že entity některých konceptů daného vztahu (závislé koncepty) jsou určeny entitami jiných konceptů (určující koncepty) (Cao, 2010, str. 11).

Funkce  $f(a_1, \dots, a_{n-1}, a_n)$  je  $u$ :

- kde  $f$  je návěští typu funkce
- kde  $u$ , konceptuální graf, je tělem definice  $f$
- kde  $u$  je alespoň  $d$  rozdílných konceptů každý asociovaný k jinému  $a_i$
- Kde existuje funkční závislost mezi hodnotami asociovanými ( $n-1$ ) prvních konceptů (identifikovaných pomocí  $a_i$  kdy  $\forall i \in [1, n-1]$ ) a hodnotu asociovanou s  $n$ -tým konceptem ( $a_n$ ). (Mineau, 1994, str. 169)

### 3.2.2 Notace

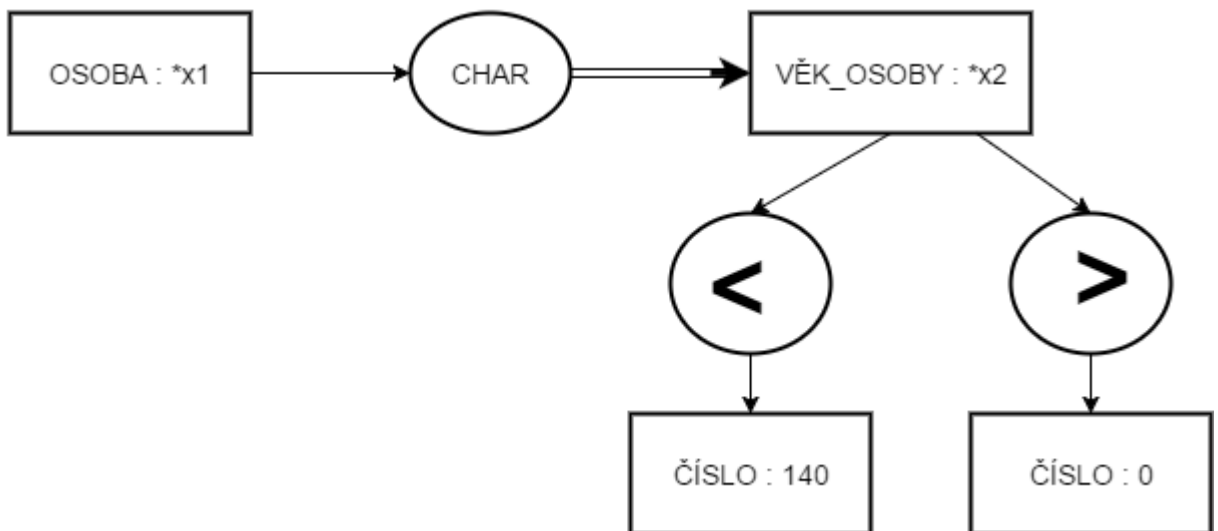
Graficky se v konceptuálních grafech znázorňují stejně jako vztahy, avšak hrana mířící od vztahu je značená dvojitou šipkou. Tato dvojitá šipka označuje poslední parametr. Funkce bude v grafické notaci konceptuálních grafů znázorněna kruhem označeným korespondujícím návěštím typu funkce  $f$ . Pokud má funkce pouze dva parametry, hrany (šipky) se nečíslují. Pokud má funkce více parametrů (o počtu  $n$ ), bude každá z  $n-1$  hran směřujících k uzlu funkce označená číslicí od  $1$  do  $n-1$  a poslední parametr je označen

dvojitou šipkou směřující od uzlu funkce. Graf na Obr. 3.1 zobrazuje funkci vztahující osobu k jejímu věku, kdy je věk osoby funkčně závislý na osobě. Jinak řečeno: (VĚK) je typem vztahu funkce se signaturou (OSOBA, ATRIBUTU\_OSOBY), [OSOBA] je určující koncept a [VĚK\_OSOBY] je závislý koncept.



Obr. 3.1 Konceptuální graf obsahující funkci (Mineau, 1994)

Funkce v konceptuálních grafech mohou být generalizovány a specializovány stejným způsobem jako vztahy. Jediným rozdílem je to, že hrana směřující od uzlu funkce je značená dvojitou šipkou. Kdyby například byla v systému již nadefinována funkce (CHAR), která identifikuje charakteristiky, které jsou funkčně závislé na zdroji konceptu, bylo by výhodné ji využít pro definici funkce (VĚK). Takovouto definici funkce zobrazuje graf na Obr. 3.2.



Obr. 3.2 Konceptuální graf definující funkci (VĚK) (Mineau, 1994)

### 3.2.3 Hierarchie typů závislostí

Cox, Delugach a Skipper (2001) ve své práci tvrdí, že pouze říci, že mezi dvěma koncepty existuje nějaký závislostní vztah nestačí, protože toto tvrzení nenese žádnou informaci. Je to vzájemně výjimečný vztah? Je závislost kauzálním vztahem? Proto navrhují hierarchii typů závislostí, která vychází z množiny atributů závislostí v daném systému

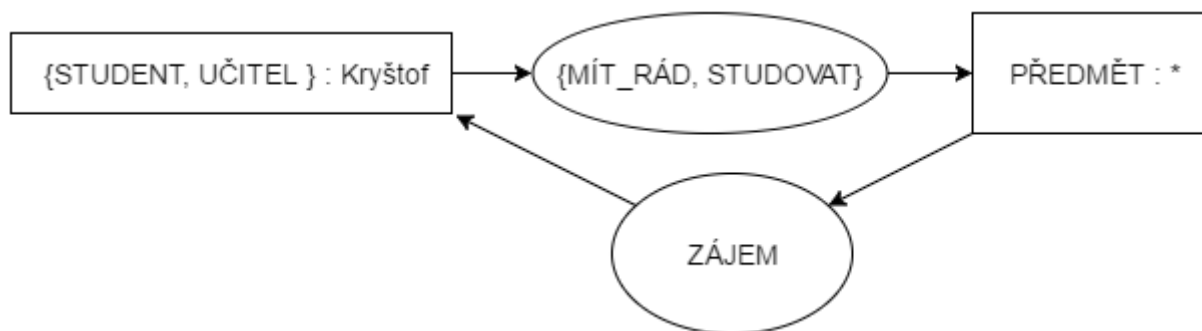
a oblasti. Tvrdí, že z takovéto množiny atributů a jejich hodnot, lze pak vytvořit částečně uspořádanou hierarchii typů závislostí, kde jsou typy závislostí charakterizovány skrze vzájemné vztahy. Ve svém článku navrhuje množinu atributů, které jsou dle jejich názoru aplikovatelné na jakékoliv do té doby známé závislosti. (Cox, Delugach a Skipper 2001)

### 3.2.4 Aktory

Aktory jsou speciálním typem konceptuálních vztahů, které do své teorie přidal John F. Sowa pro vyjádření funkčních závislostí. Sowa tedy do své teorie konceptuálních grafů zavedl tzv. funkční vztahy, což jsou typy vztahu  $t$ , které obsahují jednu nebo více funkčních závislostí. Pokud má  $t$  veličinu  $\geq 2$  může být deklarován jako funkční typ konceptuálního vztahu v jeho posledních  $m$  hranách po některá  $m < n$ . První  $n-m$  hrany se nazývají vstupní hrany a posledních  $m$  hran se nazývá výstupní hrany. Každá výstupní hrana musí být funkčně závislá na všech vstupních hranách. Formálně: pokud je  $r$  konceptuálním vztahem typu  $t$ , tak je referent každého konceptu připojeného k výstupní hraně  $r$  jednoznačně určen když jsou všechny ostatní referenty konceptů připojeny ke svým vstupním hranám specifikovány. Pokud je funkční závislost reprezentována uzlem aktoru, každá vstupní hrana vztahu je reprezentována vstupní hranou aktoru a každá výstupní hrana vztahu je reprezentována výstupní hranou aktoru. s aktory se zachází jako s jakýmkoliv jinými vztahy v konceptuálním grafu. Sowa však neformalizoval žádnou speciální notaci, ale ve svých příkladech zobrazuje aktory v kosočtvercích. (Sowa, 2000, str. 235, 2008)

### 3.3 Konjunktivní typy

Je možné říci, že v praxi dochází k tomu, že většina entit, se kterými se pracuje, spadá do více než jedné charakteristiky typu (typy mezi sebou nemají vztah nadtyp  $\geq$  podtyp). Pokud by měl být vytvořen typ pro každou individualitu, byla by nakonec množina typů příliš velká a její vytvoření velmi náročné. Navíc ne vždy by byl takový typ konceptuálně relevantní. Bierle (1992) navrhl rozdělit interpretaci svazy typů dvojitým způsobem: na svazově-teoretickou a řádově-teoretickou. s tzv. svazově-teoretickou interpretací je poslední specifický společný nadtyp dvou typů interpretován průnikem jejich interpretací. u řádově teoretické interpretace je interpretován pouze podmnožinou tohoto průniku. Bierle (1992) tvrdí, že řádově-teoretická interpretace je pro aplikace umělé inteligence příhodnější a přesnější. Graf na Obr. 3.3 reprezentuje větu: „Kryštof je student a učitel. Studuje a má rád předmět, který ho zajímá.“



**Obr. 3.3** Konceptuální graf s konjunktivním typem konceptu a vztahu (Cao, 2010, str. 14).

### 3.4 Fuzzy logika a teorie fuzzy množin

Implementace fuzzy logiky a teorie fuzzy množin je pro teorii konceptuálních grafů velmi přínosná, protože se díky ní dá reprezentovat neurčitost<sup>16</sup>. A. Zadech, autor teorie fuzzy množin, v roce 1973 popsal princip inkompatibility: „S rostoucí složitostí systému klesá naše schopnost formulovat přesné a významné vlastnosti o jeho chování, až je dosáhnuta hranice, za kterou je přesnost a relevantnost prakticky vzájemně se vylučující jevy.“ Při reprezentaci znalostí je samozřejmě velmi důležité najít rovnováhu mezi přesností a relevantností. Implementace teorie fuzzy množin to alespoň do určité míry umožňuje.

Fuzzy logiku a její využití v konceptuálních grafech zmiňoval již John F. Sowa (1984), ale nenavrhl žádnou notaci nebo operaci, jak fuzzy logiku do grafů implementovat. První, kdo aplikoval teorii fuzzy množin a fuzzy logiku byl. K. Morton (1987), který zjistil, že pomocí tehdejší podoby konceptuálních grafů není možné reprezentovat neurčitost v přirozených jazycích. Autor představoval reprezentaci nejasnosti (fuzziness) vjemů, lingvistickou a tvrzení pro typy konceptů ENTITA, ATRIBUT a INFORMACE.

Vjemová nejasnost vyjadřuje míru shody mezi reálnou entitou a typy konceptů a je vyjádřena reálnými čísly z intervalu  $\langle 0,1 \rangle$ , kdy 1 znamená úplnou shodu a 0 žádnou shodu.

<sup>16</sup> Neurčitost je jedním z charakteristických rysů expertních systémů. Zdroji neurčitosti jsou: nepřesnost, nekompletnost, nekonzistence dat, vágní pojmy, nejisté znalosti. Může se vyskytovat jak ve znalostech obsažených v bázi znalostí, tak v uživatelem zadaných faktech, která tvoří bázi dat. Vlastní povaha reality způsobuje, že poznatky, které z ní získáváme, jsou neurčité či vágní. Příčinami neurčitosti mohou být problémy s daty: např. chybějící nebo nedostupná data, nespolehlivá data (např. z důvodu chyb měření) nebo nejasná nebo nekonzistentní reprezentace dat. Druhou příčinou neurčitosti mohou být nejisté znalosti způsobené tím, že znalost nemusí být platná ve všech případech nebo může znalost obsahovat vágní pojmy. Neurčitost bývá vyjadřována pomocí jediného čísla, které je obvykle přiřazováno z intervalu  $\langle 0,1 \rangle$ , či  $\langle -1,1 \rangle$ . Existují však rovněž přístupy, v nichž je neurčitost vyjádřena intervalem hodnot, tedy dvojicí čísel, či dokonce přístupy s kvalitativním vyjádřením neurčitosti. (Dvořák, 2004, str. 8, 23)

Lingvistická nejasnost vyjadřuje vágní numerické i nenumerické charakteristiky pomocí fuzzy množin. Příkladem metrického atributu (charakteristiky) je věk osoby, který může být vyjádřen vágně jako „mladý“ nebo „ne moc mladý“. Příkladem nenumerického atributu je „barva moře“, jejíž hodnota je většinou vágní jako „modré“ nebo „zářivě modré“. Nejasnost tvrzení vyjadřuje míru shody konceptuálního grafu reprezentujícího situaci nebo tvrzení s logickou pravdivostní hodnotou pravda a je vyjádřena reálným číslem z intervalu  $(0,1)$ , kdy **1** znamená úplnou shodu a **0** žádnou shodu. Autor také rozšířil operace konceptuálních grafů, inferenční pravidlo a teorii modelů pro fuzzy konceptuální grafy. (Coe, 2010)

Wuwongse a Manzano (1993) tento první návrh fuzzy konceptuálních grafů rozšířili o fuzzy vztahy, jež nabývají hodnot z intervalu  $(0,1)$ . Wuwongse, Creasy a Cao (1997; Cao a Creasy, 2000) následně obohatili sémantiku fuzzy grafů o fuzzy pravdivostní hodnoty, díky čemuž lze mnohem lépe ve fuzzy konceptuálních grafech reprezentovat neurčitost a částečnou pravdu. Cao, Creasy a Wuwongse definovali tzv. *fuzzy typy konceptů a vztahů*, protože původní definice typů se nedaly pro fuzzy logiku plně využít. Fuzzy konceptuální graf je tedy konceptuální graf, který pracuje s fuzzy typy konceptů a vztahů. (Wuwongse, Creasy a Cao, 1997; Cao a Creasy, 2000)

### 3.4.1 Fuzzy typy konceptů a vztahů

Při snaze zařadit entity do hierarchie typů se často stává, že mají charakter neurčitosti nebo částečné pravdy. Může to být způsobeno jejich vágní přirozeností nebo nekompletností informací nutných pro přiřazení typu. Příkladem prvního jsou vágní typy konceptu [VYSOKÁ\_ŽENA] a [POHLEDNÝ\_MUŽ]. Druhý případ ilustruje sdělení někoho, kdo vidí: „Je více méně pravda, že je to PTÁK.“, kdy je tato nejistota vyjádření způsobena nějakou mírou neurčitosti v procesu vnímání (např. špatná viditelnost).

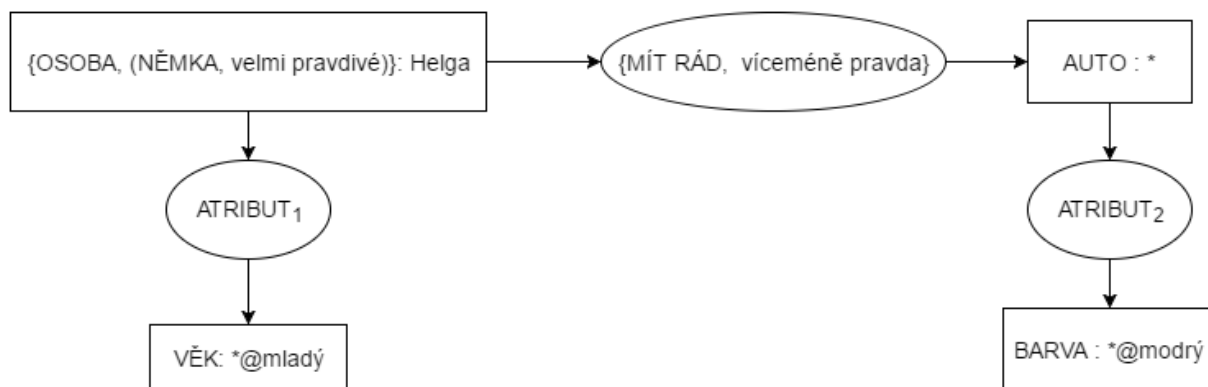
Takto špatně zařaditelnou entitu  $e$  není nutné vždy zařazovat striktně do nějakého typu  $t$ , ale je možno říci, že entita je nějakého typu  $t$  s neurčitostí a s/nebo s pravdivostní hodnotou vyjádřenou ve formě jazykové proměnné. Takový typ konceptu by vypadal následovně: [POHLEDNÝ\_MUŽ, úplná pravda]. Fuzzy typy konceptů a vztahů jsou tedy kombinace původních typů a fuzzy pravdivostní hodnotou uspořádaná do svazu. Díky tomuto kroku se dá fuzzy konceptuálními grafy provádět operace projekce, zobecnění atd. stejně jako u běžných konceptuálních grafů. i fuzzy typy mohou být konjunktivní. Stupeň

příslušnosti entity univerza  $u$  do fuzzy množiny  $F$  je dán funkcí  $\mu_F : u \rightarrow \langle 0,1 \rangle$ , (Coe, 2010, Bělohávek, 2005)

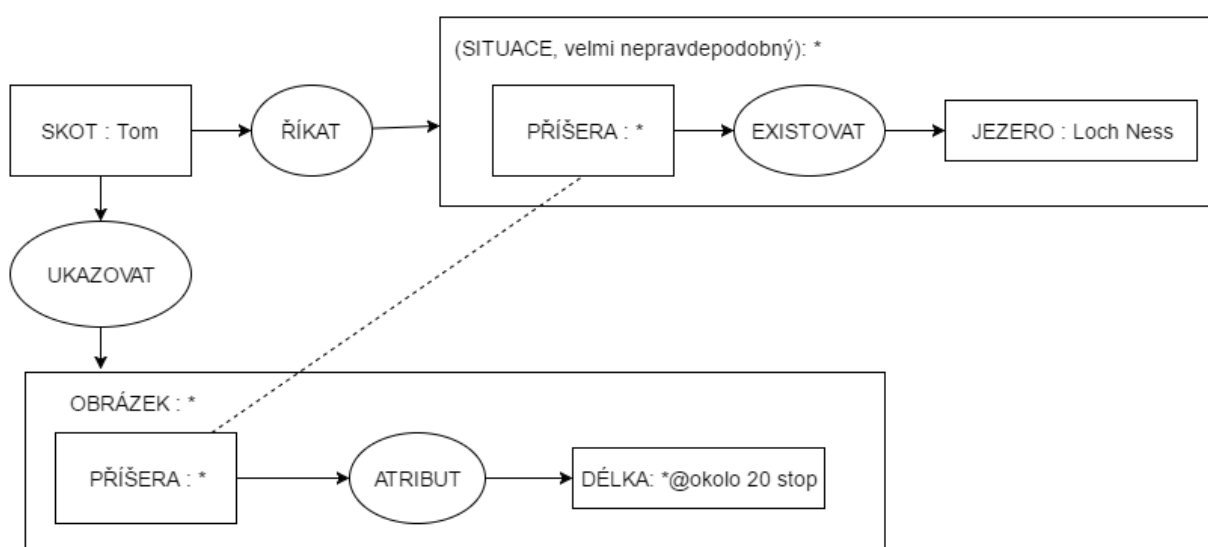
Fuzzy množina  $F$  je definována jako přiřazení, které každému prvku  $u$  univerza  $u$  přiřazuje hodnotu funkce jeho příslušnosti do fuzzy množiny  $F$  rovnou  $\mu_F(u)$ , tj.  $F = \{(\mu_F(u)) / u \in U\}$ . To znamená, že prvek patří do množiny s jistou mírou příslušnosti (stupeň příslušnosti). Funkce, která každému prvku univerza přiřadí stupeň příslušnosti, se nazývá funkce příslušnosti. Funkce příslušnosti tedy určuje hodnotu proměnných a to buď ve formě čísla z intervalu nebo jazykové proměnné. (Mařík, Štěpánková, Lažanský 2003)

Jazyková (lingvistická) proměnná je taková proměnná, jejíž hodnoty jsou výrazy nějakého jazyka. Hodnotu jazykové proměnné je možno interpretovat jako fuzzy množiny. Množina jazykových hodnot se označuje jako množina termů, jež jsou definovány na univerzu (univerzální množina). (Volná a Kotyřba 2013) Jinak řečeno fuzzy lingvistická proměnná obsahuje určitá slova (proto lingvistická proměnná) a významy těchto proměnných jsou fuzzy množiny, které jsou definovány v nějakém univerzu prostřednictvím funkce příslušnosti. v grafu na Obr. 3.4 jsou jazykové proměnné vyjadřující pravdivostní hodnotu „velmi pravdivé“ a „víceméně pravda“ a jazykové proměnné jiného typu: „modrý“ a „mladý“.

Ve fuzzy konceptuálních grafech tak krom svazu typů vztahů a konceptů existují svazy fuzzy typů konceptů a fuzzy typů vztahů. Pro definici typu je třeba fuzzy jazyková proměnná, která je definována pomocí funkce příslušnosti. Také je třeba funkce příslušnosti, která každé entitě z univerza přiřadí jazykovou proměnnou. (Cao, 2010)



Obr. 3.4 Fuzzy konceptuální graf s fuzzy konjunktivními typy pro větu: "Je velmi pravdivé, že je Helga Němka, která je mladá a je více méně pravda, že má ráda své auto, které je modré." (Cao, 2010, str. 37)



Obr. 3.5 Zahnížděný fuzzy konceptuální graf s koreferenčním spojením pro větu: "Skot Tom ukazuje obrázek příšery, která je okolo 20 stop dlouhá a vypovídá o velmi nepravděpodobné situaci, že tato příšera existuje v jezeře Loch Ness." (Cao, 2010, str. 42)

### 3.5 Reprezentace času (Časové konceptuální grafy)

P. Q. Liua a Y. Tian (2013) tvrdí, že tradiční konceptuální grafy jsou velmi komplexním nástrojem pro reprezentaci znalostí kontextu, což je nezbytné pro sémantickou a pragmatickou analýzu. Avšak poukazují na problematiku konceptuálních grafů, které mají příliš mnoho vrstev, což vzniká v situacích, kde je potřeba zahrnout příliš grafů do přílišné hloubky. k tomu nejčastěji dochází v souvislosti s potřebou reprezentovat čas, modálnost a negaci. Při porozumění během zpracování přirozeného jazyka hraje kontext významnou roli, jelikož stejné tvrzení může nabývat různých významů na základě měnícího se kontextu. v tradičním konceptuálním grafu se čas znázorňoval takto: (ČAS) →[TVRZENÍ: KONCEPTUÁLNÍ\_GRAF] . Problém nastává, když je sdělení složitější a výsledný graf je velmi komplikovaný s velkou hloubkou zahrnutí. Ve vzorci Vzor. 3.6 je zdánlivě jednoduchá věta „Neználek by nemohl letět na Mars“ reprezentována velmi komplikovaným trojvrstevným konceptuálním grafem s třemi kontexty. Takové grafy se hůře interpretují i mapují, čímž ztrácejí svoji elegantnost.

Dalším problémem je, že dva grafy, které mají různý čas nemohou být spojeny (operace *join*) a to i přes to, že jsou jejich kontexty totožné, což je samozřejmě velká překážka při odvozování. Problematické také je, že konceptuální grafy si hůře poradí s reprezentací času v jazycích, které mají víc časů. Většinou se v nich znázorňuje pouze minulost, přítomnost a budoucnost. Ale například angličtina má časů 16 a i když běžně používá pouze 8, stále je to mnoho a obyčejné označení minulosti, přítomnosti a budoucnosti nedokáže správně reprezentovat informace, které sebou nese použití jisté kombinace časů.

Liua a Tian (2013) navrhuje zapojit do konceptů znázorňující sloveso časové intervaly, aby od sebe odlišil jednotlivé časy. Výsledný konceptuální uzel korespondující se slovesem v přirozeném jazyce vypadá takto:

[ SLOVESO : (ZÁKLAD, [ZAČÁTEK, KONEC]) ]

*Vzor. 3.5 Koncept v časovém konceptuálním grafu (Liua a Tian, 2013)*



V referentu konceptu (viz Vzor. 3.5) [ZAČÁTEK,KONEC] reprezentuje časový interval, pro který platí, že  $ZAČÁTEK \leq KONEC$  a ZÁKLAD reprezentuje bod v čase řečníka.

(MINULOST) → [TVRZENÍ: (NEGACE)→

[TVRZENÍ: (MOŽNOST)→

[TVRZENÍ: [LETĚT]-

(AGENT) → [OSOBAE : Neználek]

(MÍSTO) → [PLANETA : Mars] ] ] ]

*Vzor. 3.6 Konceptuální graf věty „Neználek by nemohl letět na Mars“ (Liua a Tian, 2013)*

[ ŽÍTI : (Ted', [Ted', Ted']) ] –

(AGENT)→ [OSOBA : Maruška]

(MÍSTO)→ [ZEMĚ : Čína] ].

[ ŽÍTI : (Ted', [Ted' -4, Ted']) ] –

(AGENT)→ [OSOBA : Maruška]

(MÍSTO)→ [ZEMĚ : Čína] ].

*Vzor. 3.7 Časové konceptuální grafy reprezentující větu „Marie žije v Číně“ a „Marie žije v Číně 4 roky“ (Liua a Tian, 2013)*

### 3.6 Odvozování a konstrukce konceptuálních grafů

Konstrukce konceptuálních grafů a využití grafů pro odvozování sebou přináší několik zásadních problémů. u konstrukce je to především požadavek na graf, který skutečně odráží realitu a reprezentuje tvrzení, které je pravdivé a má smysl v přirozeném jazyce i při převodu do predikátové logiky. u odvozování je potřeba zajistit, aby si grafy reprezentující znalosti odvozené od jiných grafů reprezentující znalosti zachovaly svoji pravdivostní hodnotu a aby byl proces odvozování validní. Tato kapitola představuje především některé operace nad

konceptuálními grafy a pravidla, kterými je třeba se při tvorbě nových grafů nebo odvozování znalostí z grafů řídit.

### 3.6.1 Kánonické grafy

Konceptuální graf je kombinací uzlů konceptů a uzlů vztahů, kdy každá hrana každého konceptuálního vztahu je připojena ke konceptu. Je však zřejmé, že ne všechny takovéto kombinace dávají smysl. Někdy mohou vzniknout absurdní kombinace. Jako v příkladu ve Vzor. 3.8, kdy by konceptuální graf mohl být vyjádřen větou: „Některé akty spaní mají agenta, kterým je myšlenka, která má zelenou barvu.“ Tato věta je sémanticky správně, ale nemá žádný význam (stejně jako slavná věta Noama Chomského „Colorless green ideas sleep furiously“ (Chomsky, 1975)).

[SPÁT] → (AGNT) → [MYŠLENKA] → (BARVA) → [ZELENÁ].

*Vzor. 3.8 Konceptuální graf, který je sémanticky správně, ale nenese žádný význam (Sowa, 1984, str. 94)*

Je tedy potřeba odlišit grafy nesoucí význam, které reprezentují reálné a možné situace ve vnějším světě od těch nereálných. Pro tento účel je třeba v každém systému označit některé konceptuální grafy jako grafy kánonické. Dá se říci, že takové kánonické grafy používají i lidé v běžném životě: každý člověk si skrze zkušenost vyvinul pohled na svět reprezentovaný právě kánonickými grafy. Na základě těchto kánonických grafů je pak možno usoudit, zda je určitý graf validní nebo ne.

Jedním možným zdrojem těchto grafů je pozorování. Druhým zdrojem je derivace nových kánonických grafů pomocí pravidel tvorby. Třetím zdrojem je tzv. vhled nebo kreativita: osoba může cítit, že existující vjemy, koncepty a vztah neadekvátně popasují situaci a může vyvinout radikální novou konfiguraci, která vše lépe popíše. Vhled vede k novým kánonickým grafům, které nahradí nebo rozšíří ty staré (Sowa, 1984,3.4.)

## Kánon

Kánon reprezentuje gramatiku systému založeného na grafech. Kánon obsahuje nezbytné informace potřebné pro derivaci množiny kánonických grafů. Obsahuje slovní zásobu, která se skládá z hierarchie typů konceptuálních vztahů a konceptů a množiny individuálních označovačů. Tyto tři svazy (typů a individuálních označovačů) dohromady tvoří množinu všech individualit přítomných v systému. Kánon také obsahuje vztah, který propojuje návěští z množin typů s individuálními označovači. a v poslední řadě kánon obsahuje množinu základních grafů, které se nazývají kánonické grafy, jež kódují určitá sémantická omezení nad aplikačními doménami. (Sowa, 1984; Kent a Williams, 1996)

## Kánonická pravidla tvorby

Existují čtyři pravidla pro derivaci konceptuálního grafu  $w$  z konceptuálního grafu  $u$  a  $v$  (kdy  $u$  a  $v$  mohou být stejným grafem) nazvána kánonická pravidla tvorby. Kánonická se jmenují proto, že pomáhají odvodit každý graf od původního grafu, který se nazývá kánon (Kent a Williams, 1996). Konceptuální graf se nazývá kánonický pokud je kánodem nebo je odvozený z kánonického grafu kánonickým pravidlem tvorby. Tato čtyři pravidla se nazývají: *kopírování* (copy), *omezení* (restrict), *spojení* (join) a *zjednodušení* (simplify). Tato pravidla tvorby jsou jakousi grafovou gramatikou pro kánonické grafy. Krom definice syntax také posilují potřebná sémantická omezení. i přesto, že počet kánonických grafů může být teoreticky nekonečný, není nutné je všechny definovat, ale pravidla tvorby umožňují vytvořit pouze základní množinu kánonických grafů a ostatní z nich odvodit. Kánonická pravidla tvorby *omezení* a *spojení* jsou pravidla specializační. *Kopírování* a *zjednodušení* graf dále nespécializují, ale také ho nezobecňují.

Sowa původně zahrnul do své teorie konceptuálních grafů (1976) pravidlo *odpojení* (detach), ale později ho nahradil simplifikací. i přesto se domnívám, že je dobré na tomto místě zmínit, protože někteří autoři ve svých pracích s tímto pravidlem pracují. s pravidlem odpojení jako jedním z pravidel tvorby mohlo být deklarováno více grafů nežli se simplifikací, avšak také to znamenalo, že pravidla tvorby nebyly striktně specializačními pravidly. To v praxi znamenalo, že graf mohl nabýt po aplikaci pravidel tvorby obecnějšího charakteru, než měl před aplikací. Simplifikace sice nepřispěje k tzv. specializaci grafu, avšak také nezpůsobí jeho další zobecnění. Požadavek na to, aby pravidla tvorby měla striktní

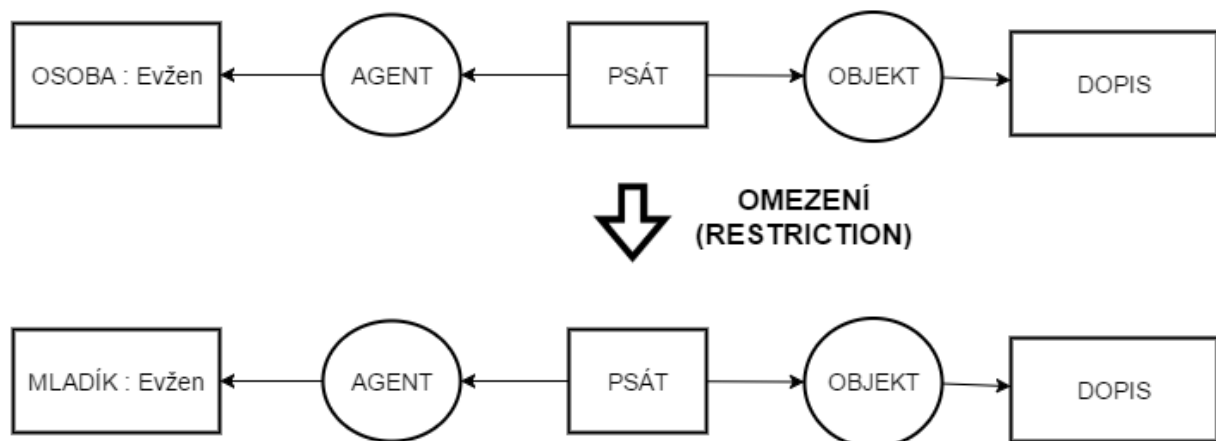
specializační charakter, byl tak silný především proto, že je potřeba pro definici zobecněné hierarchie. Současná podoba pravidel tvorby bez pravidla odpojení garantuje, že jakýkoliv graf vyplývající z jejich pravidel bude stejně nebo více specializovaný. (Way, 1994)

- **Kopírování**

Přesná kopie kánonického grafu je také kánonický graf: **w** je přesnou kopií **u**. Samozřejmě pravdivostní hodnota **u** je stejná jako **w** a model, který činí **w** pravdivou, tak bude činit i **u**.

- **Omezení:**

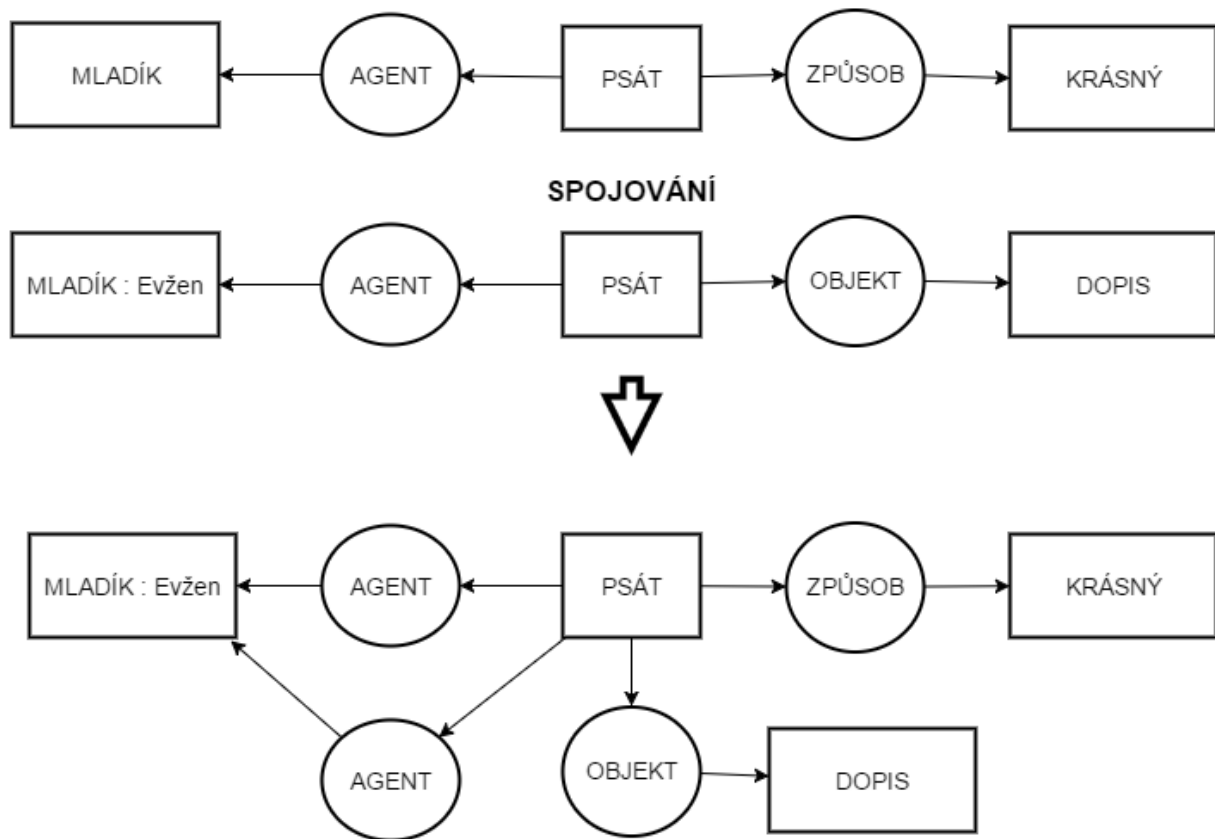
Toto pravidlo tvorby omezuje množinu možných hodnot konceptů pomocí tzv. specializace. Toho může být dosaženo dvěma způsoby. První variantou je nahrazení návěští typu konceptu návěštím podtypu jako je například odvození [DÍVKA] od [OSOBA]. Druhou variantou je konvertovat obecný koncept. Kupříkladu je možná konverze obecného konceptu [PES: \*] na individuální koncept [PES: Lassie] za předpokladu, že Lassie je v systému jasně určený pes. Pro jakýkoliv koncept **c** v u typu (**c**) platí, že může být nahrazen podtypem. Pokud je koncept **c** generický, jeho referent může být změněn na individuální označovač. Tyto změny jsou povoleny jen tehdy, pokud referent (**c**) odpovídá typu (**c**) před změnou i po ní. u individuálních konceptů je však nutné zkontrolovat shodu, aby bylo jisté, že se referent shoduje s novým návěštím typu. Takto je například možná restrikce z [PES: Lassie] na [KÓLIE: Lassie] ale nikoliv na [PUDL: Lassie], protože Lassie nebyla pudl. (Sowa, 1984)



Obr. 3.6 Pravidlo tvorby omezení (Sowa, 1984, str. 93)

- **Spojení:**

Pravidlo spojení spojuje identické koncepty. Dva grafy mohou být spojeny právě díky tomu, že se identické koncepty spojí v jediný koncept. Návěští typu a referent musejí být shodné. Koncept [OSOBA: Anička] může být spojen opět jen s konceptem [OSOBA: Anička]. Stejně jako [KOČKA] může být spojena s [KOČKA]. Pokud je jeden koncept generický (neobsahuje referent) a druhý individuální (obsahuje referent), mohou být spojeny po restrikci konceptu [OSOBA] na [OSOBA: Anička].

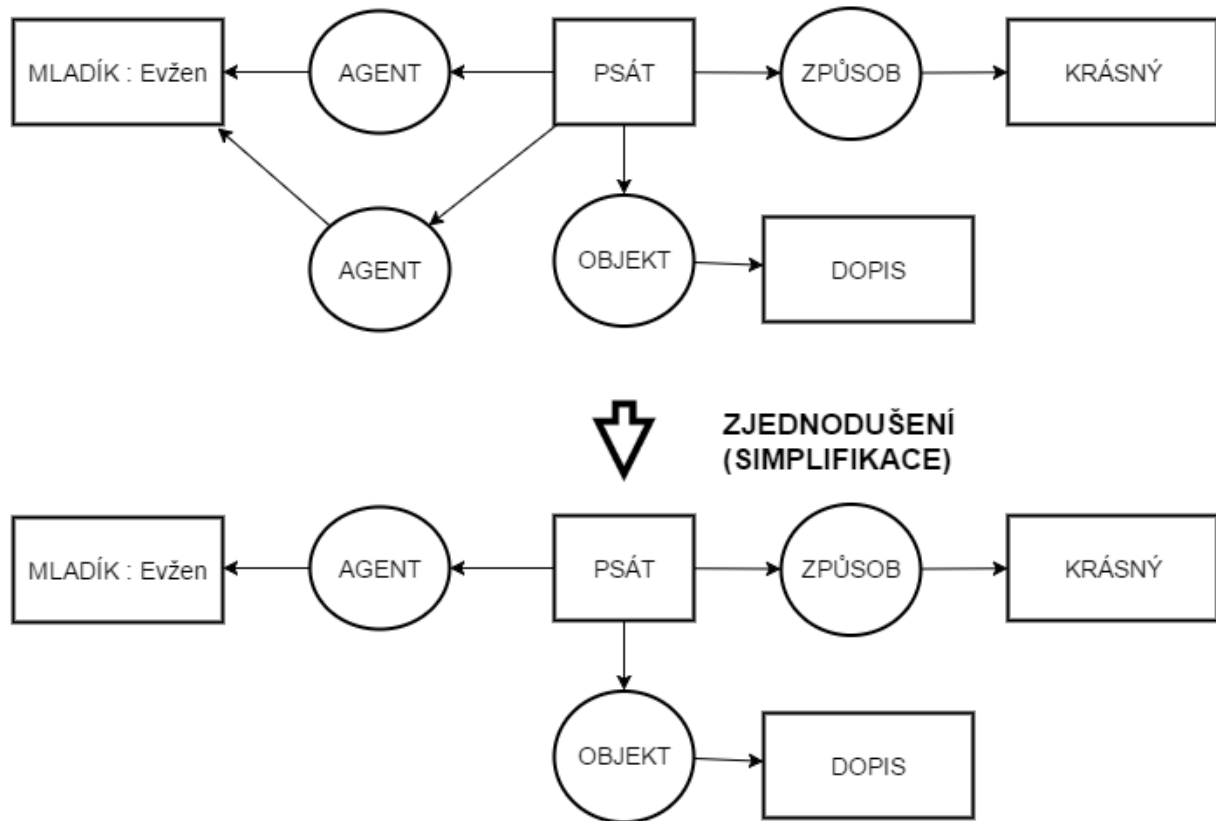


Obr. 3.7 Pravidlo tvorby spojení (Sowa, 1984, str. 93)

- **Zjednodušení:**

Pokud je koncept **c** v grafu **u** identický s konceptem **d** v grafu **v**, nechť je **w** výsledným grafem získaným vymazáním konceptu **d** a spojením konceptu **c** všemi hranami konceptuálních vztahů dříve spojených s **d**. Pokud jsou dva koncepty spojeny, některé vztahy (nebo funkce) ve výsledném grafu se mohou stát redundantní. Jeden z každého páru duplikátů může být odstraněn pomocí pravidla zjednodušení: dva vztahy stejného typu jsou spojeny se stejnými koncepty ve stejném pořadí: tudíž tvrdí stejnou informaci a jeden z nich může být proto odstraněn. Toto pravidlo koresponduje s pravidlem logiky, kdy  $R(x,y) \wedge$

$R(x,y)$  je ekvivalentem s  $R(y,x)$ . Avšak pořadí hran je důležité. Formule  $R(x,y) \wedge R(y,x)$  není ekvivalentem  $R(x,y)$ . Pokud jsou konceptuální vztahy (či funkce)  $r$  a  $s$  v grafu u duplikáty, tak může být jeden z nich odstraněn z u společně se všemi svými hranami. (Sowa, 1984)



Obr. 3.8 Pravidlo tvorby zjednodušení (Sowa, 1984, str. 94)

- **Odpojení**

Pravidlo odpojení, dovoluje odstranit konceptuální vztahy z grafu. Toto pravidlo tvrdí, že všechny souvislé grafy vzniklé odpojením konceptuálního vztahu z kánonického konceptuálního grafu jsou také validní a kánonické. (Sowa, 1976)

### 3.6.2 Základní operace zobecnění a specializace

Jak bylo řečeno, kánonická pravidla pro tvorbu grafů jsou tzv. specializačními operacemi. Někdy je však potřeba nějaký graf zobecnit. Během specializace grafů nemusí být nutně zachována pravdivostní hodnota grafu, avšak u zobecnění zachována je. (Sowa, 1984, str. 96) Každý z autorů definuje tyto operace jinak. Chein, Mugniera Croitoru (2013) tvrdí, že existuje 5 základní zobecňujících a 5 specializačních operací, protože mezi ně počítají i kopírování a zdvojení a zjednodušení. v této práci se však přikláním k výkladu Sowy (2008), který tvrdí, že tři posledně zmíněné operace jsou tzv. ekvivalenční, protože generují graf,

který je logicky ekvivalentní k originálu. Toto rozdělení se mi jeví více strukturované a názorné. Základní operace slouží k vytváření a odvozování nových grafů z již existujících a také k řešení úloh, odvozování, formulaci dotazů a k odvozování odpovědí na dotazy.

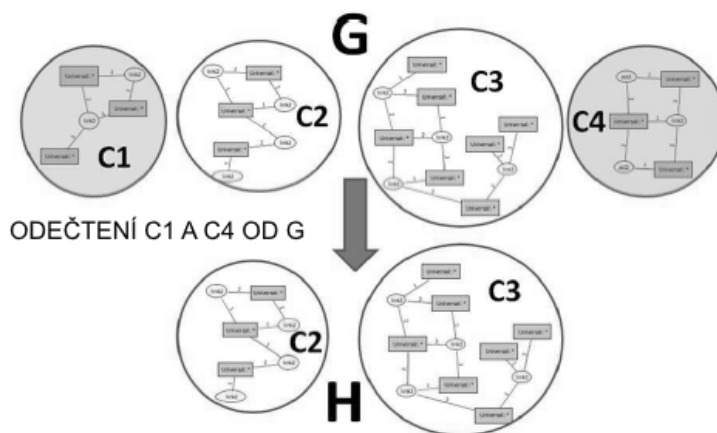
### Definice termínů „specializace“ a „zobecnění“

**Neformální definice:** Necht' je  $i$  část informace; kdykoliv je  $k$  i připojena další část informace, jedná se o specializaci  $i$  (obsahuje specifitější znalost). Naopak odstraněním části informace z  $i$  vznikne část informace, která je obecnější než  $i$  (obsahuje méně konkrétní znalost). (Chein, Mugnier a Croitoru , 2013)

**Formální definice:** Graf  $G$  je zobecněním grafu  $H$ , pokud existuje sekvence základních zobecňujících operací vedoucí od  $H$  k  $G$ . a naopak  $H$  je specializací  $G$ , pokud existuje sekvence základních specializačních operací, vedoucí od  $G$  k  $H$ . Také platí, že  $G$  je zobecněním  $G$  jenom tehdy, pokud  $H$  je specializací  $G$ . Zásadní je, že k dosažení specializace nebo zobecnění může být na grafu provedeno několik základních operací za sebou. (Sowa, 2000)

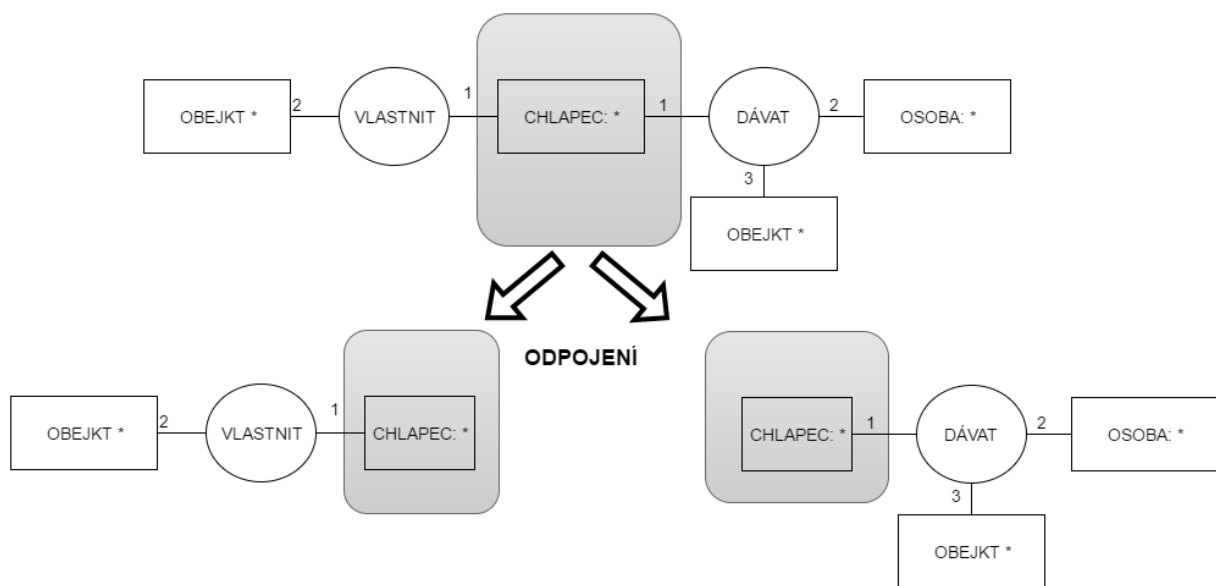
### Základní zobecňující operace

**Odečtení (Substract):** Tato operace spočívá ve vymazání některých propojených komponent konceptuálního grafu. Výsledný graf je obecnější nebo ekvivalentní původnímu grafu, protože byla vymazána část informace.



Obr. 3.9 Operace odečtení (Chein, Mugnier, Croitoru , 2013 str. 8)

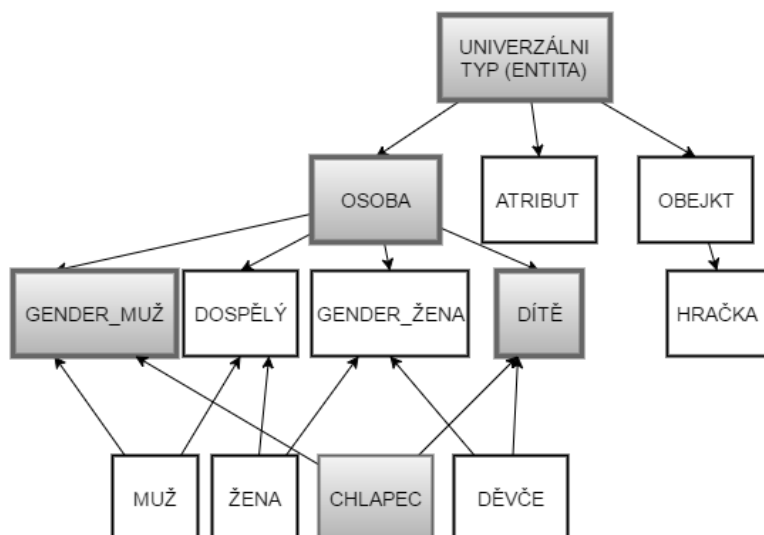
**Odpojení (Detach):** Tato operace spočívá v rozdělení uzlu konceptu  $c$  na dva uzly konceptu  $c_1$  a  $c_2$ . (Chein, Mugnier, 2009) v grafu na Obr. 3.10 je reprezentována věta: „Existuje chlapec, který něco vlastní a dává něco osobě.“ Po aplikaci operace odpojení vznikne konceptuální graf, který tvrdí, že: „Existuje chlapec, který něco vlastní a existuje chlapec, který něco dává osobě.“ Ve výsledném grafu se na rozdíl od grafu původního může jednat o dva rozdílné chlapce. Výsledný graf tedy zobrazuje obecnější situaci než graf původní.



Obr. 3.10 Operace odpojení (Chein, Mugnier, Croitoru, 2013, str. 9)

**Navýšení (Increase):** Tato operace spočívá v tzv. navýšení návěští typu konceptu nebo vztahu v určitém uzlu. Například v případě uzlu konceptu by navýšení znamenalo záměnu individuálního označovače za generický označovač, záměnu konceptu [OSOBA: Eva] za [OSOBA] nebo [MLADÍK: Evžen] za [OSOBA: Evžen]. Navýšení tedy spočívá v navýšení typu konceptu nebo vztahu na obecnější typ a/nebo jeho individuálního označovače. Tato operace je nejčastěji užívanou operací v hierarchiích. (Chein a Mugnier, 2009) Na Obr. 3.11 je vidět svaz typů, na kterém jsou vyznačeny všechny nadtypy typu [CHLAPEC], které je možné získat navýšením návěští typu konceptu.





Obr. 3.11 Svaz typů konceptu s vyznačenými variantami navýšení pro koncept [CHLAPEC] (Chein, Mugnier, Croitoru, 2013, str. 9)

### Základní specializační operace

Krom specializačních operací popsanych v kapitole, věnujících se kánonickým pravidlům tvorby; tedy *spojení* (což je opačná operace odpojení) a *omezení* (opačná operace k operaci navýšení) je definována operace *součet rozdělení* (disjoint sum)

**Součet rozdělení** (Disjoint sum): Tato funkce spojí dva rozdělené grafy pomoci toho, že vedle sebe postaví dvě kopie těchto grafů. Tato operace je opačná k operaci odečtení.

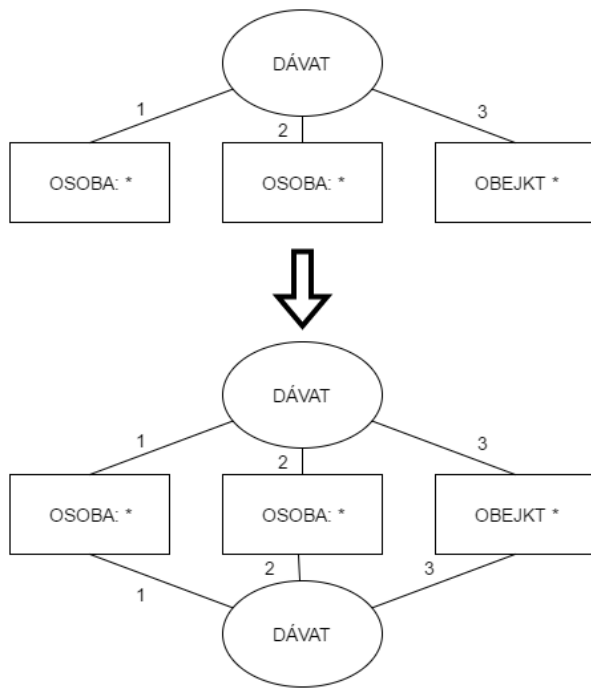
### Ekvivalenční operace

Aplikací těchto operací nedojde ke změně informací a významu v konceptuálním grafu.

**Kopírování (Copy)**: Tato operace spočívá v duplikaci celého grafu. (Chein, Mugnier, Croitoru, 2013)

**Zdvojení vztahu (Duplicate)**: Tato operace spočívá v duplikaci uzlu vztahu nebo konceptu: tedy v přidání nového uzlu vztahu, který má stejný typ a stejnou množinu argumentů jako původní uzel vztahu nebo ekvivalentního uzlu konceptu. (Chein, Mugnier, Croitoru, 2013)

**Zjednodušení vztahu a konceptu (Simplify)**: Tato operace spočívá ve vymazání jednoho z dvojice stejných uzlů vztahu. Stejná operace jde provádět na konceptech. Tato operace je opačná k operaci zdvojení. (Chein, Mugnier, Croitoru, 2013)



Obr. 3.12 Zdvojení vztahu (Chein, Mugnier, Croitoru, 2013, str. 10)

### 3.6.3 Odvozená pravidla formování grafů

Základní pravidla tvorby grafů pracují pouze s jedním konceptem, odvozená pravidla však sestávají ze sekvence **0** až **n** základních operací, které tvoří komplexní operaci. Nejdůležitějšími odvozenými pravidly formování grafu jsou: *projekce* (Projection) a *maximální spojení* (Maximal join).

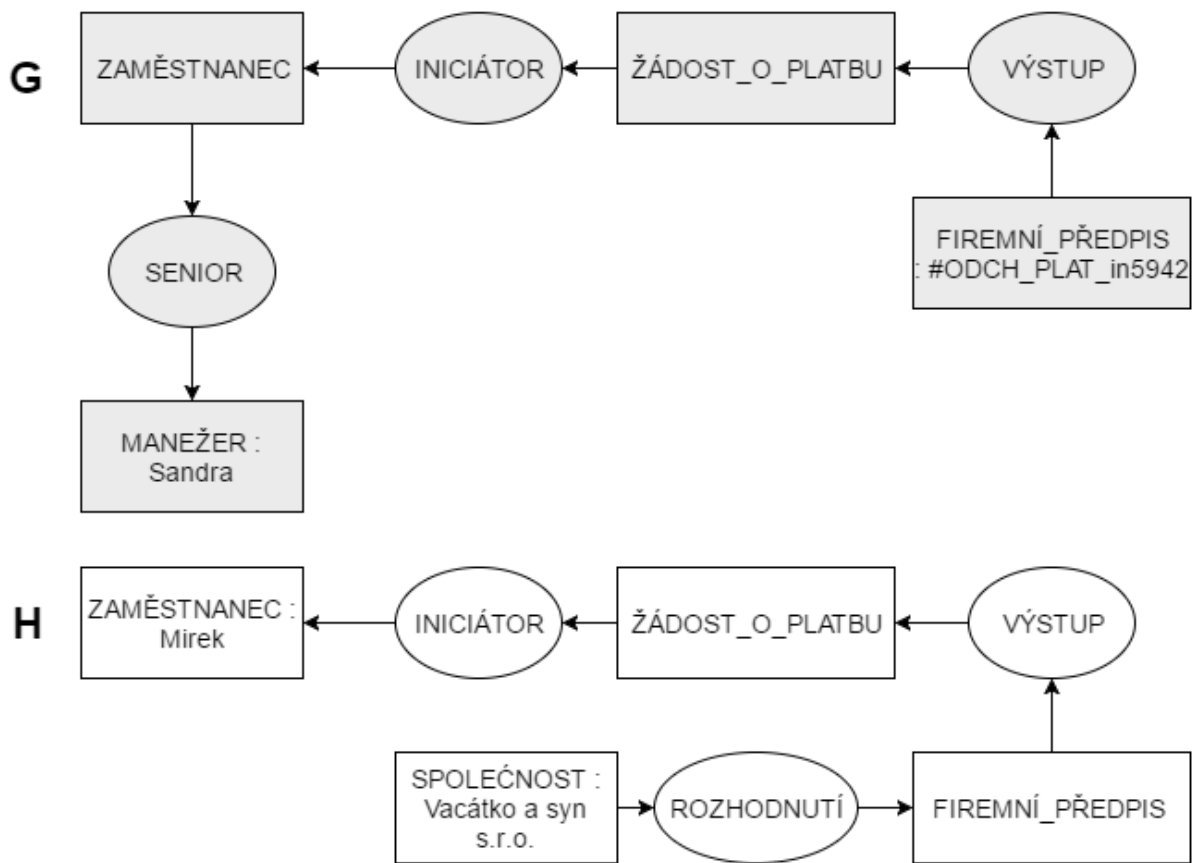
#### Projekce:

Obecně řečeno projekce je mapování z jednoho grafu do druhého. Pro každý specializovaný graf existuje obecnější graf, ze kterého byl odvozen. Stejně tak obecný graf může mít množství specializovaných variant. Projekce je tedy činnost, při které je určitý graf zkoumán, aby se zjistilo, zda neexistuje jiný graf **G**, který by byl jeho zobecněním. Pokud tomu tak je, tak je možné říci, že obecnější graf **G** se projektuje do specializovaného grafu **S** a vznikne projekce **P**. Správně zformovaný konceptuální graf je tedy projekcí jiného správně zformovaného grafu právě tehdy, když je možno definovat sekvenci **0 – n** operací omezení. Při projekci není možno využít operace spojení. Projekce může být tedy i prázdná množina

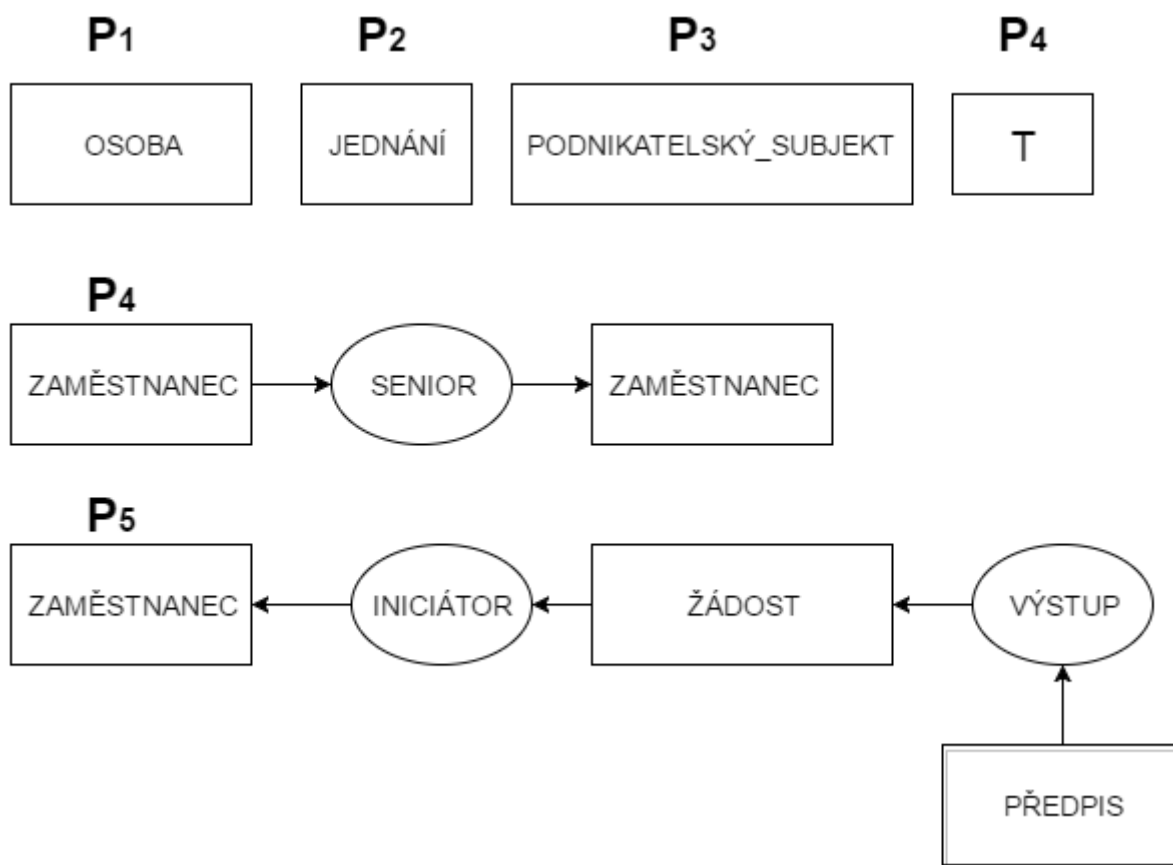
a může mít různé úrovně zobecnění. Na Obr. 3.14 jsou vyobrazeny různé úrovně projekce grafů G a H z Obr. 3.13. (Chein, Mugnier, Croitoru , 2013)

V původním grafu existuje podgraf nazývaný „původ projekce“, ve kterém každý koncept, vztah a hrana koresponduje s grafem v projekci. v grafu projekce se mohou nacházet podtypy konceptů původních grafů. Projekce a tzv. „původ projekce“ jsou navzájem izomorfní. Pokud projekce dvou grafů vytvoří identický graf, nazývá se pak společná projekce těchto grafů. Původy společné projekce v obou grafech jsou také navzájem izomorfní. Speciálním typem projekce je tzv. maximální společná projekce, která se například užívá pro operaci maximální spojení. Nejprve je třeba si určit jádro společné projekce, které je možno definovat jako množinu  $k = (a,b,c)$ , kdy  $a$  je koncept ve společné projekci,  $b$  je příslušný koncept prvního grafu a  $c$  grafu druhého. k tomuto jádru jsou následně připojovány koncepty a konceptuální vztahy. Maximální společná projekce vznikne tehdy, kdy již není možno připojit další koncept. Příklad takové maximální společné projekce je na Obr. 3.15. (Sowa, 1976)

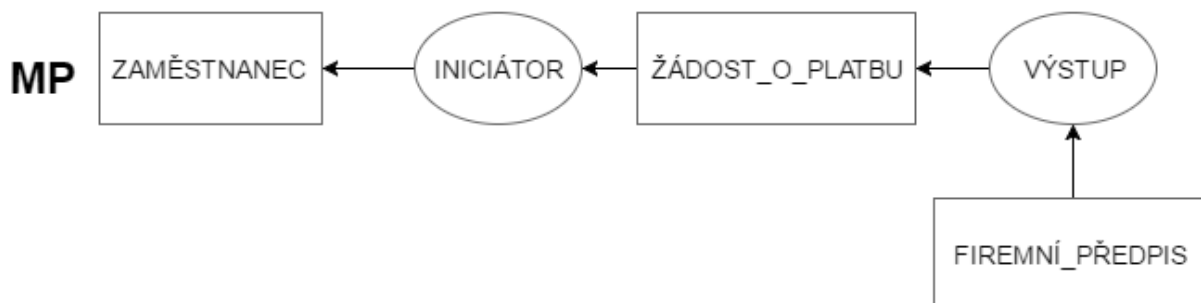
Projekce hraje zásadní roli při odvozování nových znalostí za použití konceptuálních grafů. Pokud se graf projektuje do jiného, tak může být objeven určitý vzorec. Projekce může vyústit v konstrukci nového grafu reprezentující nějaké tvrzení. Projekci je také možno využít ke kombinování grafů ve snaze vytvořit větší grafy. (Polovina, 2007)



Obr. 3.13 Konceptuální grafy G a H s různou mírou specializace a zobecnění (Polovina, 2007)



Obr. 3.14 Různé druhy projekcí grafů G a H z Obr. 3.13 (Polovina, 2007)



Obr. 3.15 Maximální společná projekce grafu G a H z Obr. 3.13 (Polovina, 2007)

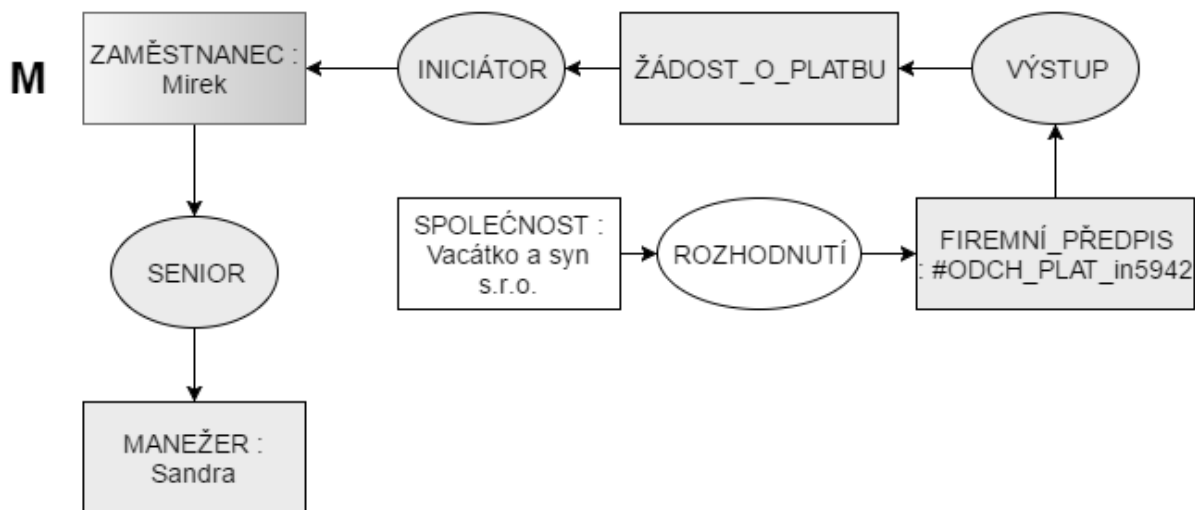
### Maximální spojení

Tato operace se používá k přesnému určení rozsahu unifikovatelného překrytí mezi dvěma konceptuálními grafy. Například při zpracování přirozeného jazyka může maximální spojení pomoci vyřešit nejasnosti a určit nejpravděpodobnější možné napojení na již existující kontext diskurzu a zpracované znalosti. (Sowa, 2008, str. 230). Maximální spojení také může sloužit k dovozování, protože do větších grafů je na nich možné uplatnit více

projekcí. o maximální spojení se jedná, když jsou grafy spojeny na nejobecnější nebo maximální společné projekci.

Maximální spojení využívá tzv. maximální společné projekce. Cílem je do jednoho grafu propojit co nejvíce dat z dvou grafů. Tam, kde to je možné, je k tomu využito například i hledání maximálního společného nadtypu dvou konceptů. Problematické je, že maximální projekce může při odvozování vést k nevalidním výsledkům. Je to proto, že Sowa ve své teorii zachází s obecným referentem jako s teoretickým ekvivalentem existenciálnímu kvantifikátoru  $\exists$  v predikátové logice. Existenciální kvantifikátor znamená: „Existuje něco, pro co...“ Například v konceptuálním grafu koncept [OSOBA], [OSOBA: \*] nebo [OSOBA: \*x] říká: „Existuje nějaká osoba, pro kterou jsou ostatní koncepty a vztahy grafu validní.“ Problém nastává právě s operací spojení, protože může spojovat koncepty různých grafů do obecného nadtypu, avšak informace v původních grafech platily pouze pro určité specifikované podtypy.

Pokud budu například tvrdit, že „Marie miluje zmrzlinu“ a „Petr nesnáší zmrzlinu“, společný nadtyp obou grafů je [OSOBA], ale spojení nemůže být validní, protože by jinak graf reprezentoval tvrzení: „Osoba miluje a nesnáší zmrzlinu“, popřípadě spojoval informace o dvou lidech pod nadtyp [OSOBA], ale tyto informace spolu nemusí vůbec souviset, protože se jednalo o rozdílné osoby. Příklad takového nevalidního odvozování je možné vidět na grafu získaném maximálním spojení na Obr. 3.16, který zobrazuje pravděpodobně nevalidní tvrzení, že Zaměstnanec Mirek je zároveň Senior manažerka Sandra. Polovina (2007) tedy navrhuje, aby se spojení užívalo jen tehdy, když jsou známy referenty, aby bylo možno předcházet takovýmto nevalidním grafům.



Obr. 3.16 Maximální spojení grafů G a H z Obr. 3.13 (Polovina, 2007)

### 3.6.4 Podpora (Support )

„Izolovaný konceptuální graf sám o sobě nemá žádný význam. Pouze skrze sémantickou síť jsou koncepty vztahy spojeny s kontexty, jazykem, emocemi a vjemy“ (Sowa, 1984). Podporu lze vnímat jako sémantiku konceptuálních grafů. Už Sowa částečně popsal některé prvky této sémantiky, ale ucelenou definici a popis obsahující veškeré prvky předložili až Chein a Mugner (1992).

Zavedení podpory do systémů umožňuje rozdělit a strukturovat znalosti do dvou rovin. První z nich je zmiňovaná podpora (někdy nazývaná podpurným slovníkem konceptuálních grafů) a lze na ní nahlížet jako na základní ontologii. Definuje vztahy mezi typy konceptů a vztahů. v druhé rovině jsou reprezentovány znalosti o vztazích mezi instancemi nebo individualitami, které jsou zakódovány ve formě základního konceptuálního grafu. Jinak řečen\ tyto základní konceptuální grafy vyjadřují fakta v bázi znalostí.

Pojem *podpora* by se dal definovat jako množina s obsahující pět podmnožin:  $S=\{T_c, T_r, M, conf, B\}$

- $T_c$  je konečná částečně uspořádaná množina typů konceptů tak, jak byla definována v kapitole 2.3.
- $T_r$  je konečná částečně uspořádaná množina typů vztahů tak, jak byla definována v kapitole 2.3.

- **M** je konečná množina individuálních označovačů, navíc existuje množina  $M \cup \{*, 0\}$  která navíc obsahuje tzv. generický označovač \* a absurdní označovač 0
- **Conf** je vztah konformity, který definuje asociační vazby mezi typem konceptu a označovačem
- **B** je množina jednoduchých hvězdovitých grafů, které u každého typu vztahu definují typy konceptů, které k němu mohou být připojeny ( určují tedy signaturu typu vztahu) (Chein a Mugner, 1992; Croitoru, Oren, Miles a Luck, 2012)



## 4 Využití konceptuálních grafů v praxi

Sowa ve své teorii konceptuálních grafů navrhl flexibilní abstraktní model, jehož možné využití je velmi široké. v průběhu uplynulých desetiletí, tak jak bylo užíváno konceptuálních grafů v praxi, se objevily rozšířené původní teorie, které v mnoha ohledech původní model vylepšily na obecné úrovni nebo adaptovaly pro potřeby dané implementace. Dalo by se říci, že témata a aplikace řešené pomocí konceptuálních grafů jsou téměř tak rozmanité jako oblast umělé inteligence.

Dotýká se oblasti zpracování přirozených jazyků (*parsování, diskurzivní analýza, generování textu, sumarizace textu, kontexty, analogie, metafora, tematické vztahy a strojové překlady atd.*), logiky a odvozování (*inferenční mechanismy, teorie dědičnosti, modalita, neurčitost, pravděpodobnostní usuzování, usuzování nad množinami atd.*), znalostního inženýrství (*akvizice znalostí, strojové učení, extrahování znalostí z textů, systémové analýzy, design databází, sémantické datové modely*), užívá se jich pro konceptuální programování, konceptuální modelování, modelování uživatelů, organizaci znalostí nebo znalostní management. a samozřejmě je možné je využít pro návrhy aplikací jako jsou expertní systémy, nástroje vyhledávání informací, inteligentní nápovědy, komunikace v distribuovaných systémech nebo počítačem integrovaná výroba. Konceptuální grafy se tedy dají využít všude tam, kde je třeba operovat z jakýmkoliv druhem znalostí, popřípadě metaznalostí.

Tato kapitola ve svém úvodu sumarizuje výhody a nevýhody a principy použití konceptuálních grafů v expertních systémech, protože to je oblast implementace, kde jsou grafy nejvíce využívány. v druhé části kapitoly je krátký přehled vybraných témat a problémů, kde vědecká obec využila konceptuálních grafů. Tento přehled zdaleka není konečný, měl by ale čtenáři přiblížit široké možnosti využití konceptuálních grafů. Ve třetí části této kapitoly předkládám ilustrativní příklad využití konceptuálních grafů v praxi pro modelování uživatele (Baldwin, Martin a Tzanavari, 2000). Tento příklad zde zveřejňuji především proto, že na něm lze velmi jednoduchým a ilustrativním způsobem popsat model fungování Inferenčního mechanismu založeného na konceptuálních grafech. Poslední podkapitola sumarizuje možnosti využití konceptuálních grafů v popisu návrhu online dostupného informačního systému určeného pro zahrádkáře.

## 4.1 Expertní systémy

„Expertní systém je počítačová aplikace nebo systém simulující poznávací a rozhodovací činnost experta při řešení složitých úloh s cílem dosáhnout ve zvolené problémové oblasti kvality rozhodování na úrovni experta. Základní součástí tvoří *báze znalostí* obsahující různé druhy znalostí a *řídící mechanismus* pro odvozování závěrů (v této práci upřednostňuji termín inferenční neboli odvozovací mechanismus), tj. program pro práci s těmito bázemi využívající technik umělé inteligence. Tyto součásti obvykle doplňuje modul pro komunikaci s uživatelem a vysvětlovací modul.“ (Kučerová, 2003)

### 4.1.1 Budování expertního systému

V první fázi budování expertních systémů je třeba nastavit specifikace systému v jazyce, který je lidem srozumitelný. Tento požadavek na srozumitelnost je dán potřebou validace obsahu výsledné báze znalostí ze strany expertu dané vědní oblasti. Takový jazyk je tzv. zprostředkující reprezentací (mediating representations), která ještě nutně nemusí být přesná, konzistentní nebo kompletní. Jejím cílem je zajistit expertům možnost volně vyjádřit znalosti a vzájemně komunikovat. Druhou fází budování je překlad co největší možné množiny neformální zprostředkující reprezentace do reprezentace formální; tedy vyjádřené formálním jazykem. Tento formální jazyk je již opatřen Inferenčním mechanismem tak, aby bylo možné ověřit, zda je výsledná reprezentace znalostí konzistentní a kompletní s ohledem na účel, ke kterému je určena. v této fázi je nejvíce problematické zajistit věrnost formální reprezentace vůči původní zprostředkující reprezentaci. Kontrola věrnosti potřebuje validaci a revizní cyklus, což je většinou dlouhý proces. Experti mohou jen s obtížemi zkontrolovat formální reprezentaci, protože málokdy rozumí jazyku formální reprezentace a Inferenčnímu mechanismu (například u lékaře se nepředpokládá znalost formálních jazyků reprezentace znalostí). Jako možné řešení navrhli Gaines a Shaw (1995) vyjádřit zprostředkující reprezentaci jazykem, který bude srozumitelný lidem a zároveň dostatečně formální pro potřeby modelování. Jak bylo již několikrát zmíněno, konceptuální grafy tyto požadavky splňují.

Vědecká obec se shoduje na tom, že jazyky reprezentace znalostí by měly mít tři základní vlastnosti: měly by být založené na logice, aby bylo možné vytvořit strukturovanou reprezentaci znalostí. Mělo by být jednoduché s nimi pracovat v počítačovém prostředí. a jak

bylo popsáno výše, pro uživatele by mělo být snadné pochopit a mít kontrolu nad každým krokem procesů v bázi. Reprezentace znalostí založená na konceptuálních grafech tyto požadavky naplňuje. Grafy jsou jednoduché matematické objekty, které mají však grafickou reprezentaci, takže je velmi snadné je vizualizovat. To umožňuje vysvětlit uživateli mechanismy odvozování, protože ty tak mohou být vizualizovány přímo skrze grafy. Druhou výhodou je fakt, že existuje velká kolekce algoritmů pracujících s grafy, což usnadňuje práci s grafy jako s počítačovými objekty. Grafy mohou být vybaveny logickou sémantikou, tudíž mechanismy založené na grafech si zachovávají kvality dané logiky, je možné s nimi provádět logické operace, a tudíž je možné využít při modelování. Konceptuální grafy jako takové jsou navíc velmi jednoduše převeditelné do logiky prvního řádu, což je pro usuzování velmi významné. (Chein, Mugnier, Croitoru, 2013)

#### 4.1.2 Báze znalostí

Základem znalostní báze je ontologie. Ontologie se dá definovat různými způsoby, ale z epistemologického<sup>17</sup> hlediska je účelem ontologie zodpovědět otázku „jaký druh věcí existuje v doméně či oblasti aplikace?“. v oblasti reprezentace znalostí je třeba mluvit o počítačových ontologiích, které poskytují symbolickou reprezentaci tříd objektů (koncepty) a možných vztahů mezi objekty (vztahy nebo role). Ontologie je tedy základním slovníkem pro reprezentaci znalostí v dané oblasti.

Znalosti jsou do konceptuálních grafů tedy v první řadě kódovány pomocí podpůrného slovníku (tzv. support vocabulary). Na slovní zásobu v tomto slovníku může být nahlíženo jako na základní ontologii skládající se z hierarchií konceptů a vztahů (popřípadě fuzzy konceptů a vztahů nebo typů závislostí atd.) Ostatní znalosti jsou v bázi vyjádřeny strukturami tvořenými termy ontologie. Jinými slovy jsou zakódovány v podobě grafů, které mají dva typy uzlů: koncepty a vztahy, respektive jsou zobrazovány pomocí instance entit a jejich vztahů, přičemž jsou tyto uzly označeny prvky z podpůrného slovníku. (Chein, Mugnier, Croitoru, 2013)

---

<sup>17</sup> „ týkající se vědeckého poznání a poznávání, též noetické“ (Klimeš, 1995, str. 157)

### 4.1.3 Inferenční mechanismus

Inferenční mechanismus pracuje se znalostmi znalostní báze za účelem zodpovědět dotaz nebo za účelem dosažení určitého cíle (například vytvořit sekvenci akcí splnění nějakého úkolu). Konceptuální grafy mají sémantiku definovanou logikou prvního řádu, což znamená, že znalostní báze může být přeložena do množin formulí logiky prvního řádu. Při řešení úkolu inferenční mechanismus však operuje přímo na znalosti definované uživatelem a ne na jejich překladech do logických formulí, což je umožněno homomorfismem<sup>18</sup> mezi grafem znalostní báze a grafem dotazu. To, že je všechny operace a úkony možné provádět rovnou na grafech a ne pomocí jazyka logiky prvního řádu, který je pro spoustu uživatelů špatně čitelný, umocňuje snadnou pochopitelnost a čitelnost Inferenčního mechanismu pro uživatele.

Aby mohly být expertní systémy používány v praxi, je nezbytné, aby uživatel chápal a kontroloval kompletně celý proces, což není jen vytvoření znalostní báze a provozování znalostního systému, ale i proces získávání výsledků či odpovědí. Uživatelem je míněn nejenom znalostní inženýr budující bázi dat ale také expert daného oboru, který ověřuje shodu implementované znalosti jeho vlastní reprezentací oblasti a také uživatel, kterému je systém určen; tedy ten, který za pomoci systému řeší nějaký problém.

Pro uživatele by tedy mělo být relativně snadné vložit do systému nějaké kusé znalosti a také se za pomoci systému dobrat k jejich pochopení. Avšak důležité je i to, aby uživatel pochopil systémem předložené závěry a odpovědi a to, jak k nim systém došel. Docílit toho, aby všichni potenciální uživatelé chápali, proč systém odpovídá určitým způsobem, je zásadní. Často to bývá také velmi komplikované, což je důsledek toho, že jsou uživatelé většinou velmi diverzní skupinou vycházející z různých vědeckých oborů a s různou mírou počítačové gramotnosti. Avšak i tento tento požadavek lze naplnit za použití konceptuálních grafů, pomocí kterých je mimo jiné i velmi snadné znalosti i proces

---

<sup>18</sup> Homomorfismus grafu  $G$  do grafu  $H$  je zobrazení  $f: v(G) \rightarrow v(H)$  s vlastností, že pro každou hranu  $xy$  grafu  $G$  je  $f(x)f(y)$  hranou grafu  $H$ . Jinými slovy graf může být vnořen v jiném. Homomorfismus zachovává cesty a tedy i souvislost grafu a komponenty. Navíc může různé komponenty propojovat. (Čada, Kaiser, Ryjáček, 2004, str.66)

Homomorfismus při projekci konceptuálních grafů způsobuje, že při projekci jednoho základního konceptuálního grafu do druhého, tedy při projekci množiny vztahů a množiny konceptů z grafu  $H$  do množiny konceptů a vztahů grafu  $G$  se zachováním hran, je možno snížit návštěví typů vztahů a konceptů v rámci hierarchie typů. (Croitoru, Oren, Miles a Luck, 2012)

odvozování vizualizovat, což může mnoha uživatelům usnadnit pochopení. (Chein, Mugnier, Croitoru, 2013)

#### 4.1.4 Příklad expertního systému v oboru zdravotnictví

Foguem, Foguem a Foguem (2014) představili beta verzi systému pro lékařské účely na bázi konceptuální grafů. Nejedná se přímo o expertní systém určený pro strategické rozhodování. Cílem je vybudovat takový systém, který shromažďuje data o pacientovi jak od lékaře, tak z lékařských přístrojů. Následně je systém schopen zanalyzovat data a navrhnout nejlepší možné řešení léčby. Systém by měl být také schopen lékaře upozornit na nové a potenciálně nebezpečné události. Především má však systém mnohem více vstupních informací než lékař. Konceptuální grafy jsou v systému použity pro reprezentaci klinických pokynů a celý systém je postaven na vizuálním odvozování. Autoři tvrdí, že vizualizace je v tomto oboru absolutně klíčová. Navíc jimi navrhnuté uživatelské rozhraní CoGui (Conceptual graph user interface) je dle autorů možné implementovat na různá pracoviště, přístroje a oddělení bez ohledu na jejich integrační platformu. (Foguem, Foguem a Foguem, 2014)

Tento navrhovaný systém se mi zdá velice zajímavý, především proto, že by v něm dle mého bylo možné zkombinovat lidské znalosti od expertů, vytěžit informační databáze a to vše pak použít v souvislosti s aktuálně získanými znalostmi od lékaře a lékařských přístrojů. Navíc by bylo možné sledovat stav pacienta v průběhu celé procedury nebo řešení nějakého zdravotního problému. Tím, že některé zdravotní potíže řeší více specialistů na vzdálených pracovištích, může leckdy dojít k opomenutí nějaké informace. Takovýto systém by dle mého přispěl k získání lepšího a celistvějšího obrazu o zdravotním stavu pacienta, čímž by bylo snadnější najít příčiny a souvislosti i zajistit správnou léčbu.

#### 4.1.5 Akvizice znalostí

Akvizice znalostí neboli získávání znalostí je proces definování pravidel a ontologií znalostních systémů. Termín byl poprvé použit v souvislosti s expertními systémy pro pojmenování první fáze budování systému; tedy vyhledání a dotazování expertů v dané oblasti a zachycení jejich znalostí ve formě pravidel a objektových a rámcově orientovaných ontologií. v dnešní době je snaha tento proces co nejvíce zautomatizovat, ale je možné říci,

že akvizice znalostí je proces získávání (buď přímo od experta nebo z jiného zdroje) informace a její formalizovanou strukturu, která umožňuje další zpracování v počítačových systémech. (Potter, 2001)

Je možné tvrdit, že většina lidských znalostí je dnes zakódována ve formě textu a fakt, že je stále více textů dostupných online otevírá nové možnosti budování znalostních systémů. Vytěžovat psané texty se tedy zdá logickou volbou. Sémantická reprezentace textů není zásadní pouze pro znalostní systémy a jejich znalostní akvizici. Může také přispět ke zlepšení uživatelské přívětivosti expertních systémů tím, že by mohla umožnit systému s uživatelem komunikovat v přirozeném jazyce. v neposlední řadě je také užitečná pro zvýšení přesnosti<sup>19</sup> v systémech vyhledávání informací. (Hensman, 2004a)

Zpracovávat však takového množství dokumentů není možné zvládnout manuálně, proto jsou pro akvizici znalostí a jejich latentní sémantiky nezbytné techniky zpracování přirozeného jazyka. Nástroje pro automatickou akvizici znalostí ještě stále nejsou dostatečně sofistikované a většina nástrojů zpracovávajících přirozený jazyk s texty zachází jako s pytlím slov (Fernández, Clergerie, Vilares, 2008). Je tedy nutné najít nástroj, který bude schopen identifikovat v textu koncepty a techniky, které si poradí s inherentní nejednoznačností a flexibilitou přirozených jazyků.

Způsoby získávání znalostí z nestrukturovaného textu se liší podle toho, zda je k dispozici definovaná ontologie dané oblasti nebo není. Zajímavou metodu zvolila Svetlana Hensman (2004a, 2004b, 2005), kdy pro vytváření konceptuálních grafů reprezentujících sémantickou strukturu textu využila existující lingvistické zdroje (VerbNet a WordNet) pro definici sémantických rolí. Metodu testovala na částech korpusu agentury Reuters a na korpusu zpráv Oddělení vyšetřování leteckých nehod irského ministerstva dopravy. a i když výsledná přesnost nebyla příliš uspokojivá, což bylo způsobeno neúplností použitých lingvistických zdrojů, domnívám se, že se jednalo o zajímavý nápad.

Ideu využít sémantické vzory z existujících lingvistických zdrojů dále rozpracovali Jiménez, Gelbukh a Sidrov (2013). Jejich cílem bylo využít konceptuální grafy pro sumarizaci textu. Nejprve je celý text reprezentován skrze konceptuální struktury, následně je

---

<sup>19</sup> Koeficient přesnosti P (Precision) je určen jako poměr počtu vybraných relevantních dokumentů k počtu všech vybraných dokumentů. Koeficient přesnosti P tedy určuje, jak dobře systém vyhledá jen relevantní dokumenty. (Červený, 1999)

ohodnocena důležitost každého uzlu konceptu a vztahu pomocí kombinace algoritmu HITS a sémantických vzorů vytěžených z databáze VerbNet. v konečné fázi jsou z konceptuálního grafu reprezentujícího text odstraněny uzly s méně důležitým obsahem, aby bylo dosaženo sumarizace grafu a tedy i textu.

Většina autorů navrhuje vytěžit lingvistickou strukturu textu pomocí robustních parsovacích<sup>20</sup> nástrojů. Ty jsou schopné z textu získat sémantické vztahy, z kterých jsou pomocí nějaké heuristické nebo statistické strategie vybrány ty nejrelevantnější. Při použití dnešních deterministických parserů je většinou počítáno s tím, že vysoká míra redundance nad velkým korpusem zamezí desinterpretaci. Fernández, Clergerie a Vilares (2008) upozorňují na to, že akvizice znalostí není deterministickým procesem, ale je nutné věnovat pozornost všem interpretacím simultánně a všechny přijatelné alternativy pak integrovat do výsledné reprezentace. Pro svůj automatický proces akvizice znalostí, kdy byla použita lingvistická analýza obsahu, klastrování a čištění dat, zvolili nedeterministický analyzátor pro tzv. „mělkou“ syntaktickou analýzu textu (shallow parsing).

Během jimi navrhnutého procesu je vytěžena struktura věty, ze které jsou získány možné hypotetické koncepty, které jsou následně ohodnoceny na základě pravděpodobnosti a vybrány ty nejdůvěryhodnější. Znalosti jsou pak mapovány do podoby konceptuálních grafů. Výsledkem je text reprezentovaný strukturou konceptů a jejich vztahů. Autoři svůj přístup vyzkoušeli na korpusu textů pojednávající o flóře Afriky. s výsledkem byli spokojeni, i když jejich nedeterministický analyzátor způsobil nadměrný výskyt neužitečných sémantických spojení, které by dle jejich názoru nebylo již tak komplikované odfiltrovat. (Fernández, Clergerie a Vilares, 2008)

Jinou techniku navrhli v roce 2012 Yang a Soo, kteří se zaměřila na automatické extrahování konceptuálních grafů z dokumentů patentových nároků. Tento směr mi přišel zajímavý, vzhledem k významu patentů v dnešním světě. Patentových dokumentů je navíc nezměrné množství a dělat například analýzu patentů pro nějakou oblast může být časově velmi náročné. Autor představuje techniku, kdy je text patentového nároku nejdříve rozdělen na množinu podvěť, které jsou dále parsovány. Takto jsou vytvořeny termy příslušející dané oblasti korpusu (např. patentová literatura z oblasti zpracování kovů). Tyto

---

<sup>20</sup> Parser je počítačový program provádějící syntaktickou analýzu.

termy jsou v textu nahrazeny unikátními symboly, čímž je zkrácená přílišná délka textu původních textů, která by mohla zkomplikovat a prodloužit sémantickou analýzu. Následně jsou zkonstruovány grafy na základě propojení předem připravené ontologie a syntaktické informace získaných termů za užití morfologické analýzy POS-tagging a závislostních stromů. (Yang a Soo,2012)

Zajímavá se mi zdála především navrhovaná ontologie, do které autoři implementoval vlastní definici typů vztahů a konceptů. Koncepty i vztahy rozdělil do dvou tříd: na závislé na zkoumané doméně (oblasti) a na ty nezávislé. Doménově nezávislé jsou například koncepty [METODA] nebo [KROK] nebo vztah (OBSAHUJE). Toto rozdělení se mi zdá vhodné proto, že když vznikne potřeba aplikovat ontologii na jinou oblast patentové literatury, postačí pouze vyměnit doménovou ontologii. Dále koncepty a vztahy rozdělil na tzv. reálné a pomocné. Reálné vyjadřují přímo obsah textu a pomocné neodráží slova původního textu, ale jsou využívány tam, kdy je nutné kupříkladu vyjádřit nějaké komplikovanější slovní spojení (například spojit dva koncepty, které tvoří jeden termín ). (Yang a Soo, 2012)

Samostatnou oblastí je akvizice znalostí ze strukturovaných zdrojů. N. Hema a S. Justus (2015) navrhuje vytěžit strukturovaný repozitář mezinárodní klasifikace nemocí ICD-10 (International Classification of Diseases). Tento repozitář je soubor sedmimístných kódů reprezentujících nemoci, zdravotní potíže, symptomy i diagnózy a lékařské procedury. Tyto kódy jsou využívány zdravotními pojišťovnami ke statistickým analýzám nejrůznějšího druhu. Autoři znázorňují klasifikaci pomocí konceptuálních grafů, protože předpokládají, že by tak mohla lépe sloužit k reprezentaci znalostí a především odvozování a její využití by mohlo být širší nežli pouze pro statistické analýzy.



## 4.2 Přehled využití konceptuálních grafů

### 4.2.1 Informační systémy

Na začátku 90. let agentura Reuters převedla svůj expertní systém CAMES (Client Admin Expert Systém) ze systému založeného na pravidlech na systém fungující na bázi konceptuálních grafů. Expertní systém zajišťoval informace finančního rázu pro brovkery, informační dealery z oblasti financí a banky. Avšak krom expertního systému poskytovala agentura svým klientům až přibližně 200 dalších produktů. Nejednalo se tedy o čistě expertní systém, ale o informační systém s mnoha dalšími moduly. Konceptuální grafy reprezentovaly kromě znalostí také znalostní schéma každého produktu, každé vlastnosti produktu i pravidla pro validace výsledků. Propojení s vnějším okolím systému (například z jiné znalostní báze) zajišťovaly uzly aktorů. Velikou výhodou oproti předchozí architektuře systému byla jeho modularita, která umožňovala snadnější rozvoj a rozšíření například v případě přidání nových produktů. (Smith, 1991)

Kirikova, Wojtkowski a Grundspenkis (2002) se pozastavují nad tím, že se konceptuálních grafů při konceptuálním modelování a při procesu návrhů informačních systémů příliš nepoužívá. Překvapuje je, že se pro modelování více využívá modelovacího jazyku UML a OCL, které jsou lidem mimo počítačový a matematický obor hůře srozumitelné. Dle počtu literatury, kterou jsem našla během rešerše se zdá, že byt se v literatuře mnoho autorů shoduje na tom, že teorie konceptuálních grafů je pro budování informačních systémů výborným nástrojem, nikdo je v praxi pro tento účel neuvžívá.

### 4.2.2 Inteligentní vyhledávání informací

„Vyhledávání informací je činnost, jejímž cílem je identifikace relevantních dokumentů nebo informací v informačních zdrojích (např. plnotextové nebo bibliografické databáze); probíhá obvykle na základě konkrétního požadavku uživatele za pomoci dotazovacích a selekčních jazyků“ (Švejda, 2003).

S informační explozí je stále zřejmější, že je nezbytné vyvinout skutečně inteligentní systém vyhledávání informací, který bude schopen vyhledávat a filtrovat informace plně automaticky a nebude s texty zacházet jako s pytlek slov. Je tedy nutné, aby takový systém pracoval s určitým stupněm porozumění. Siddiqui (2006) navrhl systém reprezentace textu

za pomoci konceptuálních grafů. Její návrh dovoluje volit míru detailnosti při zaznamenávání sémantiky do formy grafů. Tento aspekt je velmi důležitý, protože míra potřebné detailnosti se může lišit v závislosti na povaze systému a oblasti jeho užití. Autorka například tvrdí, že pro určení toho, které dokumenty jsou a které nejsou relevantní vzhledem k vyhledávacímu dotazu, není potřebná zvláště důkladná sémantická analýza (jako např. u strojových překladů, odvozování nebo sumarizace textů).

Během budování systému byl pro každou větu z textů vybraného korpusu zkonstruován konceptuální graf, který však reprezentoval jen limitované sémantické znalosti. Tyto grafy byly konstruovány na základě manuálně vytvořené částečně uspořádané hierarchie typů konceptů a množiny základních vztahů, jež vznikla analýzou syntaktických vztahů. Navrhovaný systém kombinuje statistický vektorový model, který v první fázi snížil počet vyhledaných dokumentů a konceptuální grafy, které texty roztřídí. Nakonec byl implementován i nástroj pro hodnocení relevance vyhledaných dokumentů. Při testech se systém osvědčil, i když sama autorka přiznává, že je nutné dále pracovat na zvýšení hodnocení přesnosti. Zajímavé na tomto přístupu je možnost volby míry sémantické analýzy, ale problematická se mi zdá nutnost budovat téměř celou ontologii manuálně. Siddiqui (2006) Oproti tomu například Ordoñez-Salinas a Gelbukh (2010) vnášejí do procesu budování ontologie aspekt automatizace využitím sémantických vzorů a rolí z lingvistického zdroje VerbNet. Zásadní princip vyhledávání informací za pomoci konceptuálních grafů spočívá v porovnávání grafů. Truong, Dkaki, Mothe a Charrel (2008) představili proces vyhledávání dokumentů na základě porovnání grafu reprezentujícího uživatelský dotaz a grafu reprezentující dokument. Gómez, López a Gelbukh (2001) zase uveřejnili techniku získávání sémanticky příbuzných dokumentů na základě srovnání jejich reprezentace v podobě grafu.

#### 4.2.3 Normy

Normy, neboli povinnosti, zákazy a povolení, identifikují stavy záležitostí, které by měly, mohly nebo neměly proběhnout. v umělé inteligenci se jich užívá pro regulaci činnosti tzv. inteligentních agentů, kdy normy zajišťují možnost potlačit nějaké jejich chování bez toho, aby byla potlačena autonomnost agentu. Konceptuálních grafů se dá samozřejmě využít pro vyjádření sémantiky norem, ale ukázaly se vhodné i ke zkoumání dopadu implementace norem na samotný systém. v multiagentních systémech je často aplikováno

mnoho různých norem, které ne vždy musejí být v souladu, což může vést k tomu, že interakce mezi agenty způsobí problémy v chování agentu.

Jak jsou systémy komplexnější a složitější, narůstá i počet norem a vyznat se v jejich systému i najít případné rozpory může být stále těžší. Reprezentace norem skrze konceptuální grafy může pomoci nalézt možná konfliktní místa. Je také důležité, aby mohl uživatel systému zjistit, proč se systém nějakým způsobem chová (například proč mu nedoručil potřebný dokument, nebo mu nedal odpověď na položený dotaz). Pomocí konceptuálních grafů lze jednoduše vizualizovat obsah norem, jejich vzájemné vztahy i to, které části systému jsou jakou normou ovlivněny. (Croitoru, Oren, Miles, Luck, 2012)

#### 4.2.4 Management znalostí

Konceptuálních grafů je možné využít pro management znalostí a to rovnou na několika úrovních. Krom reprezentace obsahu dokumentů, distribuce dokumentů, organizace znalostí napříč organizací, modelování uživatelů, modelování procesů, přidělování práv uživatelů atd. je lze využít například pro rozhodovací a plánovací procesy. Polovina (1997) využil konceptuální grafy pro zlepšení praktického strategického managementu rozhodovacích procesů. Zároveň výsledky jimi navrhnutého modelu porovnal s modelem založeným na kognitivních mapách, což jsou strukturované diagramy na bázi grafu. Konceptuální grafy se ukázaly jako lepší řešení, protože byly schopny zobrazit mnohem širší výsek reality. Uživatel byl tak lépe informován o aspektech a okolnostech mající vliv na řešený problém. Také uživatel získal lepší představu o tom, co ovlivnilo systém při navrhování výsledného řešení, což vedlo k možnosti lépe filtrovat chybné závěry. Tento příklad ilustruje, jak je podstatné, aby uživatel chápal procesy probíhající v systému a způsob odvozování.

#### 4.2.5 Scientometrie

Singh, Hu a Roehl (2007) publikovali výzkum, zveřejňující výsledky obsahové analýzy (kombinovali automatickou i intelektuální indexaci) dokumentů z oboru managementu lidských zdrojů v oblasti pohostinství. Cílem bylo zhodnotit uplynulých deset let vývoje této disciplíny a vytipovat hlavní proudy, témata a možné směřování tohoto oboru. Zajímavé je to, že k uspořádání znalostí získaných obsahovou analýzou použili konceptuálních grafů. Sami tvrdí, že jim tento unikátní přístup pomohl objevit a reprezentovat hlavní vývojové proudy oboru a dát je do souvislostí se státy a časovými obdobími. Tento příklad dokazuje,

že konceptuální grafy lze využít kdekoliv, kde je nutné jednoduchým a přehledným způsobem zpracovat znalosti a není to technika omezená jen na specialisty z oboru umělé inteligence.

#### 4.2.6 Sémantický web

Teorii sémantického webu jako nového evolučního stupně pojetí webu představil roku 2001 Tim Berners-Lee, tvůrce World Wide Web, ředitel konsorcia W3C (Berners-Lee, Hendler, Lassila, 2001). Bernes-Lee tehdy upozornil na fakt, že obsah internetu neustále narůstá, ale vzhledem k absenci nějaké struktury je neustále náročnější se v něm orientovat. v sémantickém webu by tedy měly být informace strukturovány a uloženy podle standardizovaných pravidel, což by usnadnilo jejich inteligentní vyhledání a zpracování. Prostředkem by měly být jazyky reprezentace znalostí schopné popisovat ontologie reprezentující oblasti zájmu. Je zřejmé, že vytvořit takto obecné ontologie je velikou výzvou, ale je pravdou, že zájem o sémantický web stojí za rozmachem používání ontologií v počítačových aplikacích.

Jako ontologické jazyky pro sémantický web byly standardizovány RDF(S)2 (Resource Description Framework Schema, neboli schéma systému popisu zdrojů), který popisuje zdroje pomocí vlastností a hodnot jejich vlastností a OWL3 (Web Ontology Language), který je oproti RDF(S) bohatší a slouží jako zápis tzv. deskripční logiky<sup>21</sup>, což z něj činí kvalitní jazyk reprezentace znalostí. Oba tyto jazyky mají ve své základní vnitřní struktuře implementovány hierarchie konceptů. (Mařík, Štěpánková, Lažinský, 2013; Rudolph, Krötzsch a Hitzler, 2007)

Rudolph, Krötzsch a Hitzler (2007) ve svém článku kritizují komunitu věnující se konceptuálním grafům, že by konceptuální grafy splňují veškeré požadavky na to stát se standardizovaným jazykem pro použití v sémantickém webu, nepodařilo se tuto variantu prosadit. Dokonce i Barnes-Lee (2001) tvrdí, že jsou konceptuální grafy použitelné pro potřeby sémantického webu a to pouze s malou modifikací, která by každému konceptu a každému vztahu přidělila URI. Sám přiznává, že při vývoji teorie sémantického webu o konceptuálních grafech neuvažoval, protože se původně domníval, že jsou využitelné spíše pro zpracovávání přirozených jazyků.

---

<sup>21</sup> Deskripční logika je skupina příbuzných logických formalismů, jejichž konstrukty slouží primárně pro formální popis konceptů a relací (snaha zachytit strukturu tříd a relací).

Každopádně užití konceptuálních grafů v prostředí sémantického webu je stále možné. Vzhledem k tomu, že je možné konceptuální grafy mapovat do jazyku standardizovaných W3C (Yao, Eitzkorn, 2006), je možné je stále používat pro reprezentaci znalostí například uvnitř znalostní báze a komunikaci s okolím zajistit standardizovanými jazyky. Zajímavý je také projekt WebKB, který užívá konceptuálních grafů pro reprezentaci sémantického obsahu webových dokumentů. Zároveň pak tyto informace v podobě konceptuálních grafů vkládá do HTML kódů stránek (Studer, 2007). WebKB sice využívá již nedefinované ontologie, jedná se však o manuální akvizici znalostí a pro širší použití tak není použitelný.

Jak lze tedy konceptuálních grafů využít v prostředí sémantického webu? Foguem, Foguem, Foguem (2014) se rozhodli využít konceptuálních grafů kvůli jejich snadné vizualizaci a mapovatelnosti do OWL a RDF(S)2. Dobrev a Strupchanska (2005) navrhli anotaci webových stránek sémantického webu za pomoci konceptuálních grafů. Tvrdí, že díky jednoduché vizualizaci bude reprezentace sémantického obsahu stránek čitelnější pro konečné uživatele. Dieng-Kuntz a Corby (2006) svůj nástroj sémantického vyhledávání CORESE (který byl prakticky využit pro aplikace znalostního managementu) rozhodli rovnou vystavět na kombinaci RDF(S) a konceptuálních grafů.

Osobně se domnívám, že zkombinovat jednotlivé jazyky reprezentace znalostí je možná tou ideální cestou. Raimbault (2012) v rámci případové studie porovnal jazyky OWL, konceptuální grafy a UML pro reprezentaci znalostí a odvozování. Zatímco UML má vysoké parametry expresivnosti a je dobře čitelný pro uživatele, nedá se využívat pro Inferenční mechanismus. Naopak OWL je pro běžného uživatele nečitelný. Konceptuální grafy mají mezi těmito třemi jazyky nejhorší expresivnost, ale jsou lidsky čitelné a ukázaly se býti nejlepším nástrojem pro zodpovídání dotazů. Autor navrhuje kombinaci těchto jazyků takovým způsobem, aby bylo využito jejich předností v co největší míře.

#### 4.2.7 Modelování a modelování uživatele

Konceptuální grafy lze využít jako modelovací jazyk použitelný pro jakékoliv modelování a to včetně modelování při vývoji nového softwaru, nebo budování informačního systému. Díky implementaci funkcí lze v konceptuálních grafech znázornit závislosti, charakterizovat jejich typy a atribut, což je nezbytné pro závislostní analýzu potřebnou pro jakýkoliv proces modelování. Příkladem využití konceptuálních grafů může

být i reprezentace znalostí o uživateli pro potřebu modelování uživatelů. Příklad modelování uživatelů určitého systému je uveden v kapitole 4.3. Avšak objevily se i snahy vytvořit takový model uživatele, který by mohl být využíván zároveň několika systémy.

Tiroshi (2012) upozorňuje na stále zvyšující se přetížení aplikací služeb a webových stránek, jež je vyvolané všudypřítomnou konektivitou a tvrdí, že tento problém je možné řešit lepší personalizací. Vzhledem k požadavku na implementaci modelu uživatele do každé služby a k faktu, že informace o uživatelích jsou uloženy v nejrůznějších formátech napříč širokým spektrem různých systémů (online služby jako email, bankovníctví, zdravotní péče, e-obchodování a samozřejmě sociální sítě) vzniká požadavek na vytvoření interoperabilního modelu uživatele. Vývoji takového modelu stojí v cestě krom komerční konkurence také absence standardu pro uchovávání modelů uživatelů. Autor navrhuje model založený na konceptuálních grafech, který však splňuje požadavek na interoperabilitu i charakteristiky potřebné pro pokročilé modelování uživatele. Zajímavé je, že se grafů pro modelování uživatelů příliš neužívá (kromě přístupů založených na ontologiích).

Tiroshi (2012) navrhuje nejdříve vytvořit základní reprezentaci znalostí o uživatelích založených na grafech pomocí extrahování z již existujících ontologií a sémantických sítí vytvořených pro reprezentaci modelu uživatele. Následně by pomocí nějakého relevantního nástroje přiřadil hodnotu jednotlivým uzlům, nebo by byl každý element grafu ohodnocen dle relevance pro uživatele.

Základní myšlenka je dle mého rozhodně dobrá, ale autor nepředkládá žádné reálné řešení a jeho návrh přináší více otázek než odpovědí. Nepředkládá žádný faktický návrh grafu a jím navrhovaný způsob vyhledávání informací není příliš komplexní ani podrobný. (Tiroshi, 2012) Paradoxně se mi tak jeví mnohem starší koncept (Baldwin, Trevor a Tzanavari, 2000) úplnější a s minimálními úpravami a za použití modernějších technologií umělé inteligence a získávání dat o uživatelích by dle mého mohl být využit nejen jako základ standardu, ale mohl by splňovat i požadavek interoperability, protože by každý systém mohl použít jen tu část grafu, která je pro něj užitečná. Ovšem domnívám se, že získat informace o uživatelích (a autor navrhuje především informace ze sociálních sítí) by bylo extrémně náročné.

Avšak domnívám se, že zde vyvstává jiná závažná otázka. i přesto, že modelování uživatelů vnímám jako velmi důležité téma, obávám se, že komplexní reprezentace znalostí

o uživateli může být mnohými vnímána jako extrémní a nepříjemný zásah do soukromí. Nemyslím si, že jsem ochotná riskovat tak velký zásah do mého soukromí ve prospěch většího pohodlí mého a především poskytovatelů služeb, kterým by univerzálně přenosný model uživatele poskytl krom jiného samozřejmě i vyšší zisky. Také si dovoluji tvrdit, že takto úplné a přenosné modely uživatelů by mohly být snadno zneužitelným nástrojem. Avšak samozřejmě tato volba leží na každém uživateli zvláště a věřím, že navýšení uživatelské přívětivosti, kterou by zavedení takového modelu přineslo, by mnoho lidí ocenilo.

#### **4.3 Příklad z praxe: využití konceptuálních grafů pro funkci inteligentních agentů pro modelování uživatelů**

Informační technologie využívá stále širší spektrum uživatelů (v porovnání s minulostí, kdy se softwarovými systémy pracovali jen specializovaní, velmi motivovaní odborníci). Nárůst počtu uživatelů sebou přináší i větší diverzifikaci skupiny, která sebou přináší rozšíření možných uživatelských potřeb, schopností, preferencí a zájmů. Výsledkem se stal logický požadavek, aby se informační systémy staly oproti minulosti více individualizované a vznikla snaha zaměřit se na naplnění rozdílných uživatelských požadavků. S touto relativně nedávno vzniklou situací se zvyšují požadavky na implementaci technologií pro modelování uživatelů přímo do informačních systémů.

V současné době se stále zvyšuje požadavek na implementaci modelu uživatele do informačních systémů, sociálních sítí, aplikací atd. Většinou však není možné, aby byl proces modelování uživatelů dlouhý a příliš podrobný a málokdy je k dispozici velké množství relevantních dat. Naopak se často vychází z malého množství dat, které je nutné shromáždit rychle, takže jsou sbírána během krátkého časového úseku, i když jsou často postupně doplňována. Tento fakt je logickou příčinou toho, že není možno využít model vyžadující přílišné množství dat. Je možné tvrdit, že lidé často podobnou techniku uplatňují i v běžném životě. Například při prvním setkání mají lidé tendenci očekávat od druhé osoby modely chování, které si vyvodili z modelu vytvořeného na základě předchozích zkušeností. Tudíž často lidé očekávají od osoby pracující v určitém oboru nebo studující nějaký vědní obor určité spektrum znalostí či schopností.

#### 4.3.1 Modelování uživatele

Modelování uživatele je proces, při kterém řešitel či designér vyvíjí kognitivní model lidského uživatele, který obsahuje i informace o jeho schopnostech a deklarativních znalostech<sup>22</sup> Většina interaktivních systémů obsahuje, i když jen nepřímo, nějaký model uživatele, se kterým přichází do interakce (Rich, 1979). Na pojem „model uživatele“ se dá nahlížet v různých perspektivách. Model uživatele může být specifický modul interaktivního systému obsahující vědomosti ohledně základních znalostí uživatele, jeho záměrech a plánech ve vztahu k úloze. Popřípadě může odkazovat na styl interakce, kterým rozhraní disponuje, bez ohledu na konkrétní uživatelské znalosti. Druhým typem jsou tzv. „mentální modely uživatele“, což jsou struktury poznatků, které si lidé vytvářejí za účelem pochopení a vysvětlení svých zkušeností a vnitřní reprezentace informací. Nebo také můžeme říci, že to je model představ, který si uživatel vytváří o daném počítačovém systému (Allen 1990). Zájem o modelování uživatele pramení z možnosti, že modelování uživatele zlepší kolaborativní funkce interaktivních systémů (Fisher, 2001). (Ferbarová, 2010)

#### 4.3.2 Inteligentní agenty

Existuje samozřejmě mnoho odlišných způsobů modelování uživatelů a i rozdílné softwarové aplikace nebo uživatelská rozhraní vyžadují jiný přístup. Jednou možností je využití tzv. inteligentních agentů. v oblasti umělé inteligence se agenty dají definovat jako jednoduché počítačové systémy, které jsou schopny autonomně<sup>23</sup> vykonávat akce v rámci svého prostředí, aby naplnily určité cíle. Agent povětšinou vnímá své prostředí pomocí sensorů (fyzické senzory pro agenty umístěné v reálném světě nebo softwarové senzory v případě softwarových agentů) a mají k dispozici repertoár akcí, které mohou vykonat k dosažení modifikace svého okolí a často se řídí nedeterministickým algoritmem. Obyčejný agent třeba vnímá teplotu v pokoji, a když je příliš nízká, tak zapne topení. Inteligentní agenty nejen že umí reagovat na své okolí, ale umějí být i proaktivní, tudíž umějí převzít iniciativu, aby dosáhly svého vytyčeného cíle. a v neposlední řadě mají sociální schopnost,

---

<sup>22</sup> Autorský překlad angl. pojmu „Declarative Knowledge“: znalost faktické informace, obsahující epizodickou i sémantickou složku. Podle Sternberga znalost a pochopení faktických informací o objektech, myšlenkách a událostech v prostředí („vědět něco“, nikoli „vědět jak“) (Kognitivní server, 2010)

<sup>23</sup> Autonomie je zde vnímána jako schopnost agentů jednat bez intervence člověka nebo jiného systému, kdy mají agenty kontrolu nad svým vnitřním stavem a jednáním. (Wooldridge, 1995)



kdy jsou schopny vstupovat do interakce s jinými agenty a lidmi, aby naplnily vytyčené cíle. (Wooldridge, 1995, str. 4-11).

Inteligentní agent sloužící k modelování uživatele, který bývá přítomný v moderních systémech, je možno nazvat tzv. osobním asistentem, který se dívá uživateli přes rameno a poznává a učí se uživatelské charakteristiky, aby mohl jednat v jeho zájmu a pomoci mu. Uživatelské charakteristiky specifikované v modelu uživatele je vhodné zkombinovat s vhodnou bází znalostí, kterou inteligentní agent využije k odhadnutí lidského chování a potřeb a reakcí a na základě toho pomůže uživateli uspokojit jeho potřeby, poradí při plnění úlohy a pomůže využít systém efektivněji. Je však pravdou, že jak ukazuje současná podoba například sociálních sítí, kde se modelování uživatele také používá, lze podobný model využít k cílení reklam a jiným ne až tak prospěšným aktivitám. Dle mého stojí za povšimnutí podobnost agentů s expertními systémy, kdy však expertní systémy na rozdíl od agentů většinou nezískávají vstupní informace ze svého okolí, ale spíše skrze uživatele. Stejně tak nepůsobí přímo na své okolí (ať již softwarové nebo reálného světa), ale častěji poskytují výstupy a rady třetí straně. Přesto se dle mého jedná o velmi příbuzné téma.

#### 4.3.3 Teoretický základ a východiska

V následujících odstavcích se pokusím představit, kterak vědci James F. Baldwin, Trevor P. Martin a Aimilia Tzanavari (2000) ve své studii navrhli využití konceptuálních grafů při konstrukci inteligentních agentů pro modelování uživatelů. Můžeme říci, že navrhli nástroj reprezentace znalostí o uživateli. Jak bylo řečeno, existují různé způsoby, jak získávat informace o uživateli, na základě kterých jde uživatel následně modelovat. Avšak většinou je k dispozici velmi omezený soubor potřebných informací, protože takovéto informace není lehké získat a navíc to vyžaduje čas. v rámci procesu modelování se objevují dva zásadní problémy. První z nich souvisí s reprezentací znalosti modelu uživatele. Vzniká otázka, jak model zkonstruovat, jak využít model v kombinaci s dalšími znalostními bázemi inteligentního agenta tak, aby byly kvalitně odvozeny důsledky a rozhodnutí. Druhý problém souvisí s problematikou fúze odlišností.

Studie (Baldwin, Trevor a Tzanavari, 2000) neřeší způsob a proces získávání informací o uživateli systému nebo softwaru. Avšak tvrdí, že když jsou sesbírána data o uživateli založená na jejich interakci s počítačem, je možno utvořit skupiny neboli klastry typů uživatelů rozdělené na základě jejich potřeb, schopností a chování. Vzniknou tak obecné

kategorie lidí s podobným chováním a charakteristikami. Tyto charakteristiky jsou zachyceny pomocí jednotlivých definicí klastrů. Každý klaster je definován pomocí relevantní fuzzy množiny atributů a vztahů (relací) mezi nimi a je reprezentován konceptuálním grafem. Následně je možno využít těchto klastrů jako tzv. prototypů. Studie tedy předpokládá již existenci nedefinovaných prototypů uživatelů.

Existuje několik možností jak prototypy nebo řekněme klastery v systému reprezentovat. Například je možno každého uživatele reprezentovat jako jeden bod vektoru a následně najít průměrný vektor v rámci klasteru, který by odpovídal prototypu. Následně by stačilo každého případného uživatele propojit s nejbližším vektorem klasteru a tím ho zařadit. Nevýhodou této metody je přílišné zjednodušení reprezentace, jelikož neumožňuje brát v úvahu vztahy mezi koncepty, ale jedná pouze na základě základních atributů. Tato metoda také postrádá flexibilitu, jelikož prototyp bude reprezentován pouze jedním bodem. Konceptuální grafy jsou naopak dostatečně flexibilní a mohou zobrazovat klaster jako množinu atributů a jejich vztahů. Konceptuální grafy také umožňují popisovat atributy jako fuzzy množiny a vztahy mezi nimi. Při užití vektorového modelu může dle mého názoru dojít ke ztrátě informací a souvislostí, které se mohou plně vyjevit až teprve ve chvíli, kdy jsou data a informace o uživatelích uspořádána ve formě konceptuálních grafů.

Následuje fáze, kdy je možné přiřadit již určitého uživatele (například Petra) vhodnému prototypu. První možností je dát samotnému uživateli vybrat, do jaké kategorie patří. Tento postup sebou nese jistá úskalí (přeceňování či podceňování se ze strany uživatele či fakt, že může uživatel spatřovat tento proces jako zdržení a obtěžování). Další možností je uživateli položit nějakou množinu dotazů a na základě jeho voleb či odpovědí ho zařadit do předdefinovaného klasteru.

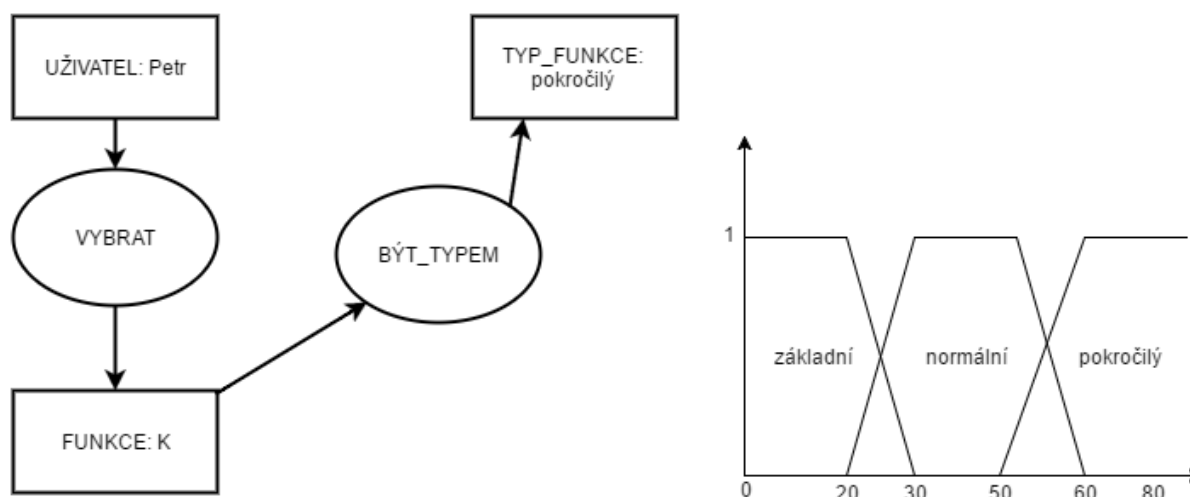
Poslední možnost, kdy si potřebné informace získá sám provozovatel systému, vyžaduje od uživatele nejmenší úsilí, čímž ho nejméně obtěžuje a je také možné ji považovat za nejpokročilejší. Otázkou zůstává, jak takovéto informace získat. Stále je asi možno považovat za nejlepší způsob pozorovat ho při práci se systémem v reálném čase. Je možné se tak dozvědět mnoho o jeho zájmech, preferencích, zvycích atd. Navíc to, že mnoho systémů poskytuje technologie uživatelského profilování a tím dávají svým uživatelům jistý stupeň svobody, podílí se tím uživatel na vylepšování inteligentního agenta. Dále je vhodné,

aby měl uživatel kdykoliv možnost podávat zpětnou vazbu a vylepšit svůj profil. Získané informace o uživateli budou následně reprezentovány pomocí konceptuálního grafu uživatele.

Nový uživatel bude porovnán s prototypem a bude rozhodnuto, do jaké míry se shodují. Tento uživatel bude porovnán i se všemi ostatními prototypy, které obsahují informace, jež jsou předmětem zájmu. Lze předpokládat, že uživatel, který bude odpovídat modelům chování vlastním určitému prototypu, bude mít také ostatní vlastnosti tohoto prototypu. i v případě, že nebude mít systém všechny potřebné informace o specifickém uživateli, bude je tak schopen odvozovat na základě prototypů uživatelů s podobným modelem chování. Je však třeba upozornit na to, že tato forma odvozování bude mít induktivní nebo dokonce analogický charakter a nepředkládá žádné záruky pravdivosti. Tato metoda nabízí možnost v první řadě třídít uživatele dle jejich charakteristik a především odvozovat informace o specifickém uživateli, které systém nemá k dispozici.

### Konceptuální grafy

Návrh pracuje s množinou konceptuálních grafů prototypů a druhou množinou konceptuálních grafů reprezentujících informace o specifických uživateli. Konceptuální grafy v tomto případě obsahují uzly vztahů, které mají jen návěští a referenty uzlů konceptů jsou instancí konceptu, jež může nabývat hodnoty, množiny hodnot nebo fuzzy množiny.



Obr. 4.1 Konceptuální graf reprezentující větu „Petr si vybírá pokročilé funkce.“ (Baldwin, Trevor a Tzanavari, 2000)

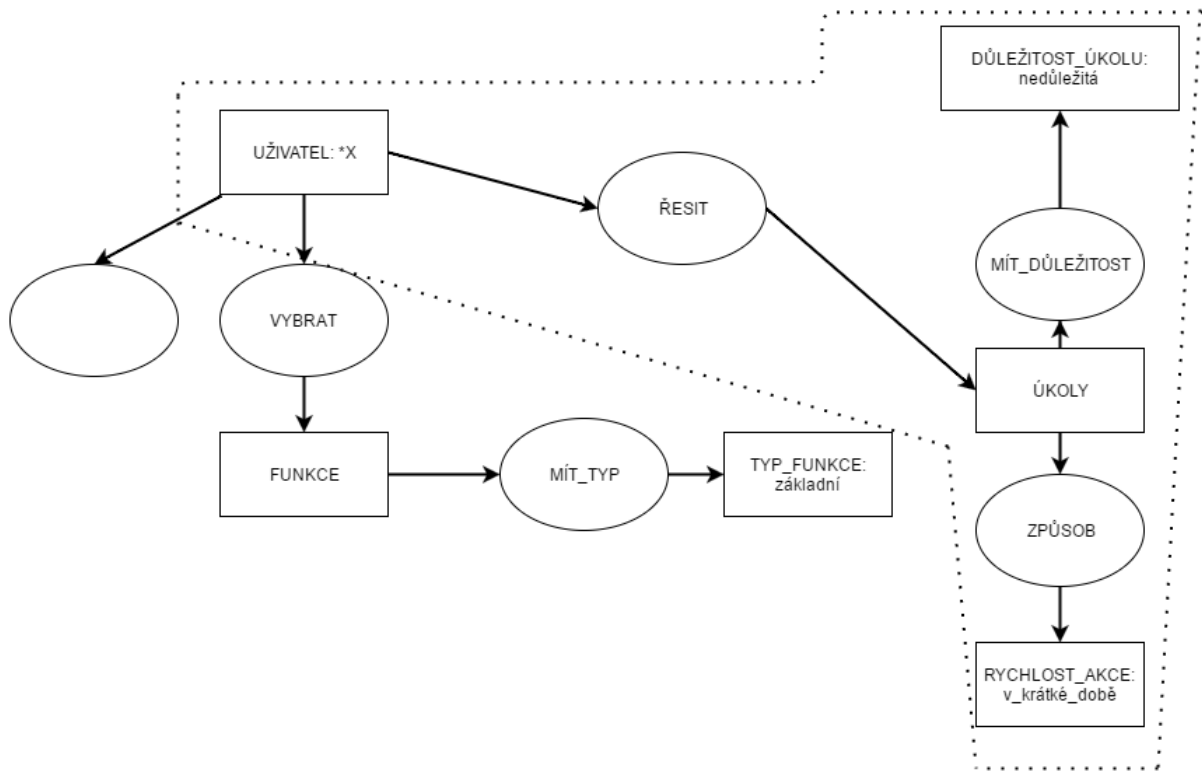
Koncepty v grafu na **Obr. 4.1** [UŽIVATEL] a [FUNKCE] jsou instancemi pro jednotlivé hodnoty „Petr“ a „K“, zatímco koncept [TYP\_FUNKCE] má referenta fuzzy množinu „pokročilý“, která je definována za pomoci fuzzy logiky. Uzly vztahů (VYBRAT) a (BÝT\_TYPEM) jsou propojeny s výše popsánymi koncepty.

#### 4.3.4 Inferenční mechanismus<sup>24</sup>

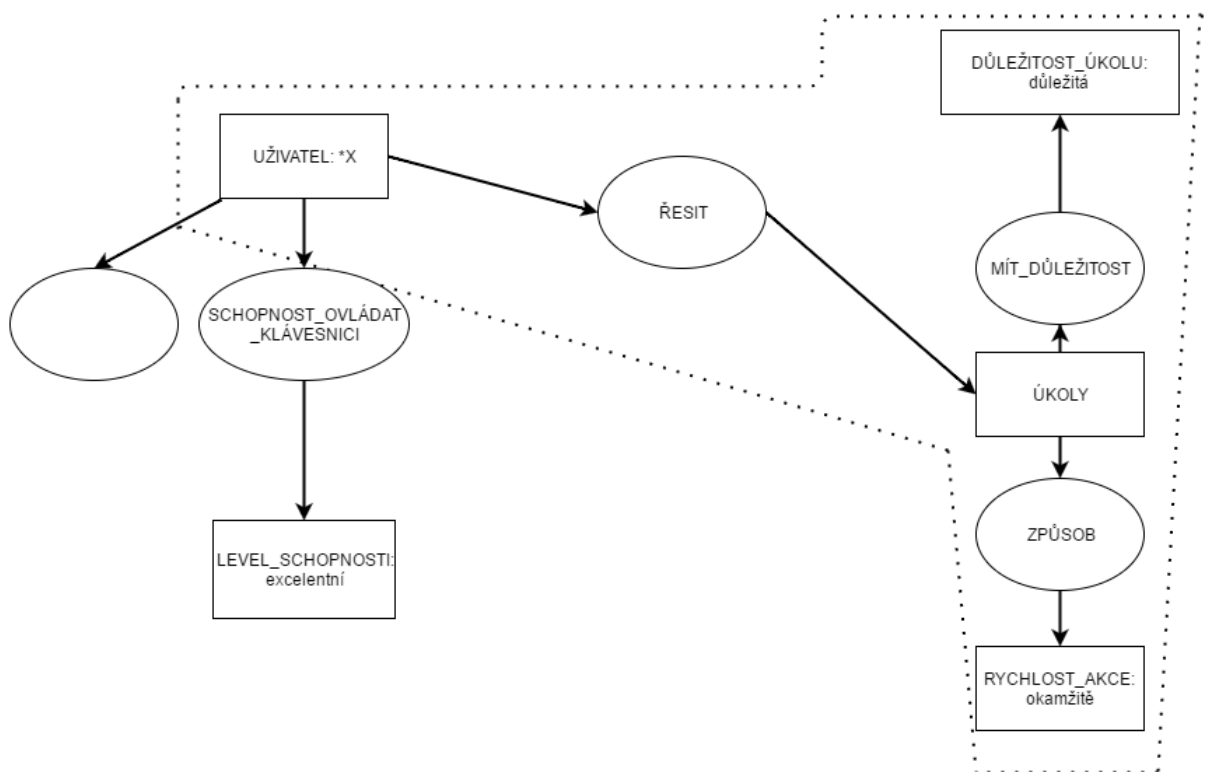
Existuje předpoklad, že jsou již shromážděna data o uživateli založená na jejich interakci s počítačem, na základě kterých jsou uživatelé rozděleni do klastrů a tak je vytvořena množina konceptuálních grafů prototypů {P1, P2}. Také je již vytvořen konceptuální graf s určitým uživatelem jménem Petr na základě získaných relevantních dat o jeho osobě, potřebách, preferencích, chování atd. v reálném systému by existovalo samozřejmě mnohem více grafů prototypů, ale pro potřeby této práce budou dva příklady dostačující.

---

<sup>24</sup> Inferenční mechanismus je jedním ze základních komponent expertního systému. Jedná se odvozovací algoritmus, vlastní program, který s využitím báze znalostí modifikuje bázi faktů, až nalezne požadované řešení úlohy. Typický inferenční mechanismus je založen na: inferenčním pravidle pro odvozování nových poznatků z existujících znalostí a strategii prohledávání báze znalostí. (Volná, Kotyrba, 2013)

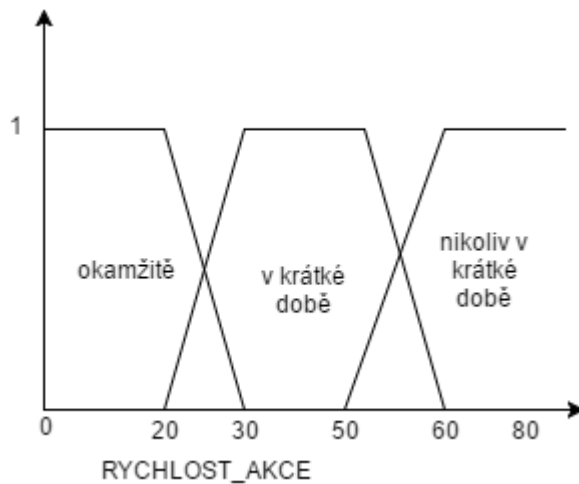


Obr. 4.2 Konceptuální graf prototypu P1 a P'1( ohraničený přerušovanou čarou) (Baldwin, Trevor a Tzanavari, 2000)

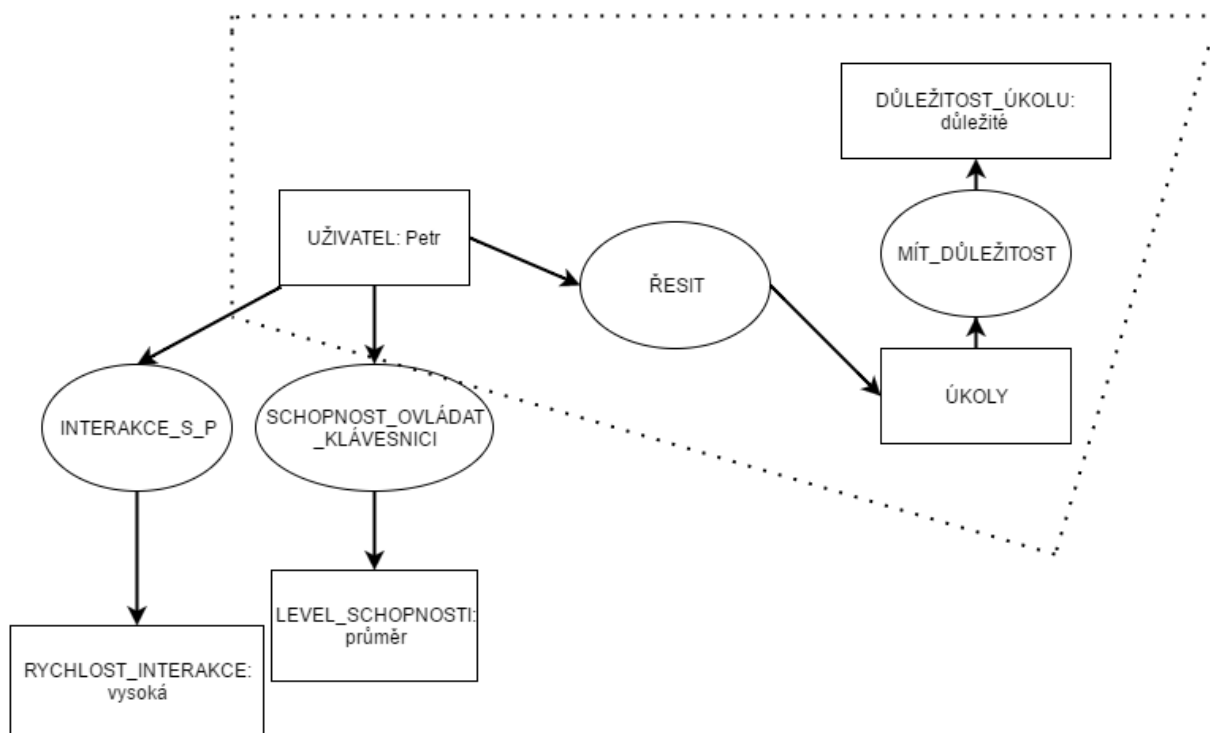


Obr. 4.3 Konceptuální graf prototypu P2 a P'2(ohraničený přerušovanou čarou) (Baldwin, Trevor a Tzanavari, 2000)

(Baldwin, Trevor a Tzanavari, 2000)



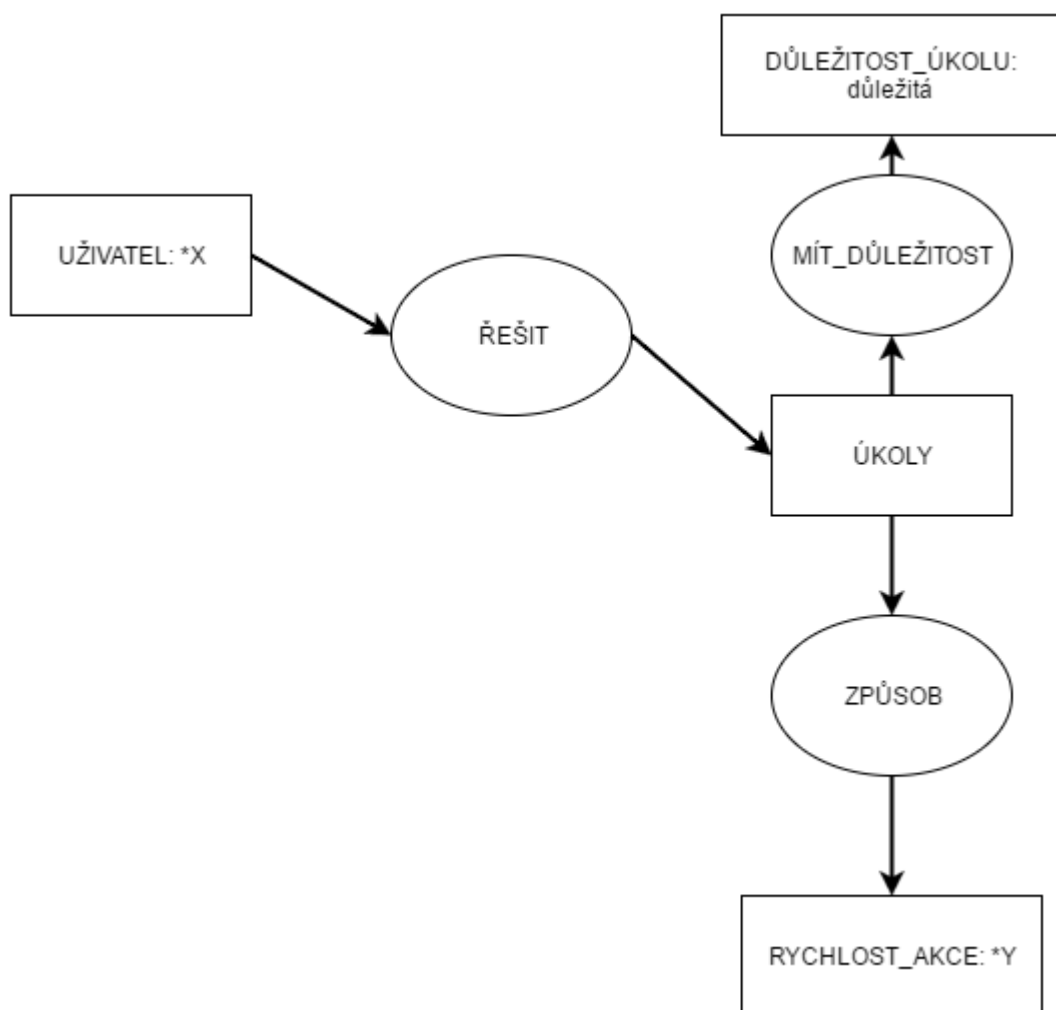
Obr. 4.4 Funkce definující fuzzy množinu obsaženou v referentu konceptu [RYCHLOST\_AKCE] (Baldwin, Trevor a Tzanavari, 2000)



Obr. 4.5 Konceptuální graf s znázorňující model uživatele Petr (Baldwin, Trevor a Tzanavari, 2000)

Jak bylo zmíněno výše, systém často nemá o určitém uživateli všechny nezbytné informace, které potřebuje, aby mohl přijmout vhodná opatření směřující například k lepšímu uspokojení potřeb uživatele. Jelikož jsou znalosti systému limitovány, je nutné vyvodit závěry z množiny dostupných informací. Kdyby graf uživatele odpovídal grafu

prototypu, daly by se odpovědi získat z části příslušného grafu prototypu. Informace, které bude systém hledat (respektive chybějící v grafu uživatele) je třeba vyjádřit za pomoci dotazovacího formátu a to kupříkladu ve formě přirozeného jazyka. Příkladem takového dotazu vyjádřeného v přirozeném jazyce je věta: „Jak rychle uživatel jedná, když řeší důležitý úkol?“ Vzhledem k tomu, že věty v přirozeném jazyce lze reprezentovat za pomoci konceptuálního grafu, vznikne tak konceptuální graf **Q** reprezentující dotaz na chybějící informaci o uživateli. Pro zkonstruování grafu **Q** je třeba použít uzly konceptu a vztahy vyskytující se v nějakém z existujících prototypů, aby mohlo být dosaženo nějaké shody. v tomto případě se bude jednat o uzel konceptu s prázdným referentem RYCHLOST\_AKCE.



Obr. 4.6 Konceptuální graf **Q** reprezentující dotaz: „Jak rychle uživatel jedná, když řeší důležitý úkol?“ (Baldwin, Trevor a Tzanavari, 2000)

V další fázi je třeba porovnat **Q** s každým grafem z množiny **{P1, P2...Pn}** (tato operace odpovídá operaci maximální spojení) a uzly grafu **Q**, které se shodují s nějakým

uzlem v grafech z množiny  $\{P_1, P_2 \dots P_n\}$  se propojí. Pokud není nalezena žádná shoda mezi uzly v grafech  $Q$  a  $\{P_1, P_2 \dots P_n\}$ , tak na dotaz nelze zodpovědět. Po provedení operace maximální spojení vzniknou grafy, ve kterých budou vytvořeny koncepty s prázdnými referenty. Budou tedy nalezeny části grafů, které odpovídají odpovědi na dotaz, a zbytek bude odtržen. Tyto grafy budou vypadat jako  $P'1$  a  $P'2$  (ohraničené přerušovanou čarou) a budou označeny prvky množiny  $\{P'1, P'2 \dots P'n\}$ . Stejným způsobem jsou provedeny operace na grafu s reprezentujícím uživatele, kdy vznikne graf  $S'$  zbavený nerelevantních uzlů.

Vzniklý graf  $S'$ , který vypadá jako graf ohraničený přerušovanou čarou na Obr. 4.5. Tento graf bude navíc obsahovat koncept [RYCHLOST\_AKCE] s prázdným polem referenta. Následně je třeba použít operaci maximální spojení a postupně porovnat  $S'$  se všemi grafy z množiny  $\{P'1, P'2 \dots P'n\}$ . Výsledkem budou grafy z množiny  $\{P''1, P''2 \dots P''n\}$ . Porovnáváním dvou grafů je míněno porovnání jejich uzlů vtaů a konceptů. Dva konceptuální vztahy si odpovídají, pokud mají stejný typ a dva koncepty si odpovídají s podporou (support) = 1, když jsou jejich typy a referenty identické a s podporou <1, pokud jsou jejich typy konceptu stejné, ale referenty se liší. Vzhledem k tomu, že referenty mohou být fuzzy množiny, tak v těchto případech je třeba užít tzv. operace Point Semantic Unification (Baldwin, 1993) aby byla získána hodnota podpory. z každé dvojice odpovídajících konceptů je získána podpora a celková podpora bude tedy získána pomocí konjunkce dílčích podpor. Výsledkem bude množina podpor  $\{s_1, s_2 \dots s_n\}$  pro každou dvojici  $S'$  a  $P'n'$ . v příkladu předkládaném v této práci budou generovány tyto informace: z  $P''1$ : „Petr řeší nedůležité úkoly v krátké době.“ a z  $P''2$  „Petr řeší důležité úkoly okamžitě.“.

Dalším krokem je uchopit části grafů  $P''1, P''2 \dots P''n$ , které odpovídají hledané odpovědi. Existuje tedy množina grafů odpovědí  $\{A_1, A_2 \dots A_n\}$ . Tyto grafy odpovědí je možno získat projekcí grafů  $P''1, P''2 \dots P''n$  na graf dotazu  $Q$  a následnou identifikací části původního grafu, která se na nic neprojektuje. v předkládaném případě budou grafy odpovědi obsahovat pouze koncept [RYCHLOST\_AKCE] s referentem odpovídajícím hodnotě relevantní fuzzy množiny (viz Obr. 4.4)

V této fázi tedy existuje množina grafů odpovědí  $\{A_1, A_2 \dots A_n\}$ , kdy je pro každý prvek z této množiny známá hodnota podpory. Pro získání finálního grafu odpovědi a je třeba



provést fúzi, kterou autoři provádějí za pomoci speciální metody vyvinuté jedním z autorů (Baldwin, 1993) a ve které mají odpovědi podobu fuzzy množin s připojenými podporami. Výsledný graf odpovědi a bude obsahovat informace, které systém potřebuje, aby mohl provést akce ku prospěchu uživatele. Rozhodně by bylo vhodné implementovat možnost úpravy a vylepšení finálního odvozovacího procesu výsledného grafu a na základě zpětné vazby od uživatele.

#### 4.3.5 Příklad odvozování v aplikaci Forum

Baldwin, Trevor a Tzanavari (2000) navrhli prototypové konceptuální grafy pro aplikaci Forum. Forum byla aplikace, jež nabízela virtuální kolaborativní pracovní prostředí, poskytující lidem, kteří se nemohou potkat v reálném světě, prostor pro setkávání a spolupráci (McGrath, 1998). v dnešní době, kdy existují velmi sofistikovaná řešení v Cloudu se tento systém jeví značně jednoduše a zastarale. Avšak domnívám se, že o to lépe na něm půjde ilustrovat použití konceptuálních grafů jako reprezentace znalostí o uživateli.

Forum bylo rozděleno do dvou částí. Contact Space (prostor kontaktování) a Meeting Space (místo setkávání). Během své studie se Baldwin, Trevor a Tzanavari (2000) zaměřili na Contact Place, které sestávalo z počtu zón, jež reprezentovaly předměty zájmu uživatelů. Na základě hlavního předmětu zájmu uživatele byl jeho avatar umístěn společně s avatary jiných uživatelů v příslušné zóně, kde mohou chatovat. Cílem nebylo pouze zařadit každého uživatele do kategorie a následně příslušné zóny. Cílem bylo především vytvořit inteligentní systém, který dokáže odvozovat na základě informací, které má k dispozici, ale i těch chybějících a kromě zařazení uživatele do fóra také umí spojovat uživatele s podobnými zájmy. (Baldwin, Trevor a Tzanavari, 2000)

Cílem bylo vytvořit rozdílné kategorie uživatelů, které by byly reprezentovány množinou prototypních modelů uživatelů. Na základě hledání podobností mezi získanými modely uživatelů by bylo možné určit, jaké typy informací (konceptů a vztahů) by prototypy měly obsahovat, protože to lze často určit až na základě analýzy získaných informací a dle zaměření systému či aplikace. Autoři upozorňují na nezbytnost získávání informací z různých zdrojů, aby nebylo opomenuto nic důležitého. Avšak také zdůrazňují nutnost vyvarovat se přílišného množství detailů, které by snižovaly flexibilitu systému. Informace je vhodné uspořádat takovým způsobem, aby se daly snadno rozlišit jednotlivé prototypy. Autoři

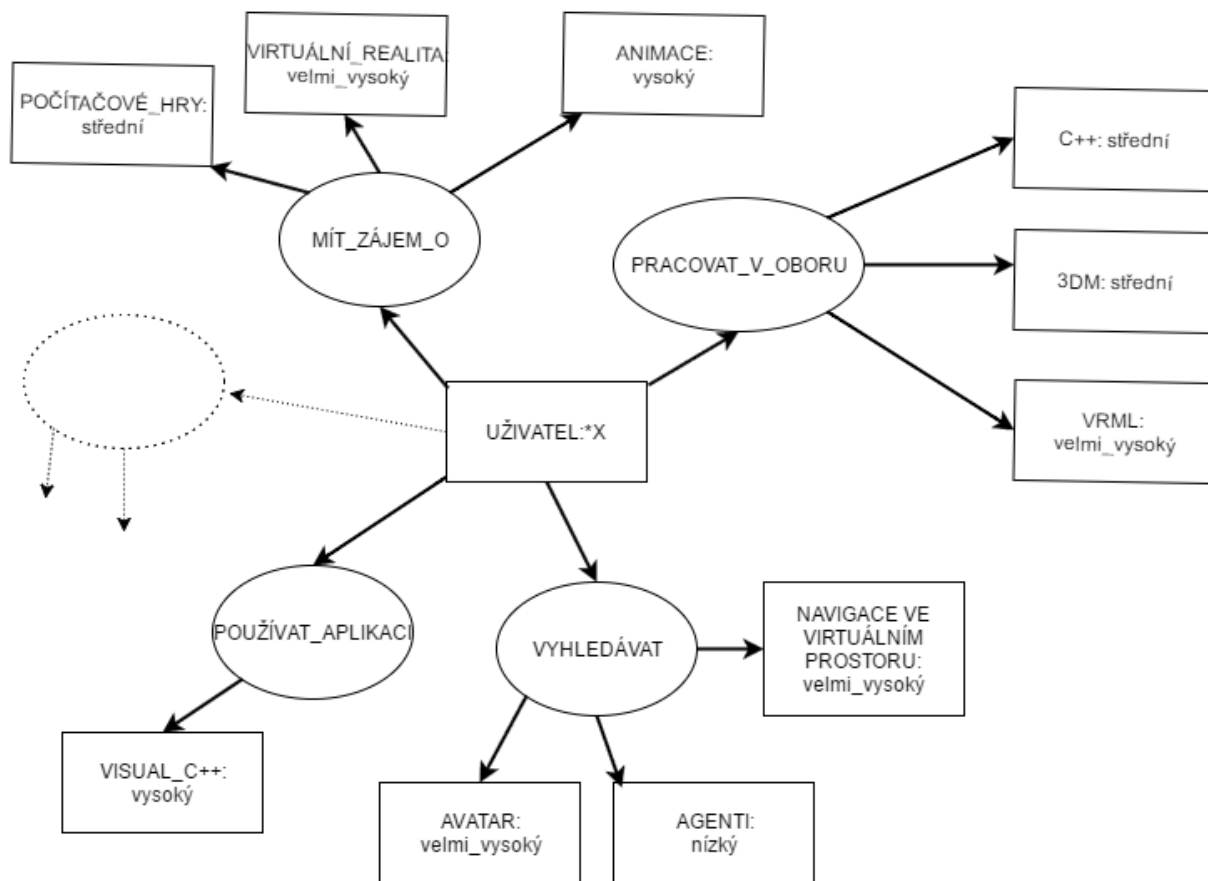
navrhují možné zdroje informací o uživateli: jedná se o to, v jaké oblasti uživatel pracuje, jaká témata vyhledává na internetu, jaké aplikace používá, jaké jsou jeho zájmy. (Baldwin, Trevor a Tzanavari, 2000) Dle mého názoru, pokud budou předem dobře nadefinovány hierarchie typů konceptu, nebude již komplikované sesbíraná data sumarizovat do formy konceptuálních grafů rozdílné úrovně konkrétnosti.

V navrhovaném systému (Baldwin, Trevor a Tzanavari, 2000) měl každý prototyp odpovídat jedné zóně Fóra. Jedná se o velmi striktní rozdělení, které by dle mého názoru již nevyhovovalo požadavkům, protože důraz na individuální personalizaci systému je ještě větší než před 15 lety. Pro zjednodušení je však vhodné si ukázat příklad s tehdy navrhovaným systémem. Je tedy možno předpokládat existenci prototypů odpovídající zónám Fóra: Softwarové inženýrství, Sdílené virtuální světy, Sémantický web atd. Každý z těchto prototypů obsahuje charakteristiky, které jsou v každém případě vlastní osobám zabývajícím se daným tématem. Například osoba, jež by v každém případě spadala do kategorie „Sdílené virtuální světy“ by měla předměty zájmu: virtuální realita, animace a počítačové hry, uměla by pracovat s VRML a C++, užívala aplikace jako je Visual C++ a The Forum a na internetu by vyhledávala hesla jako je „avatar“ a „navigace ve virtuálním prostoru“.<sup>25</sup>

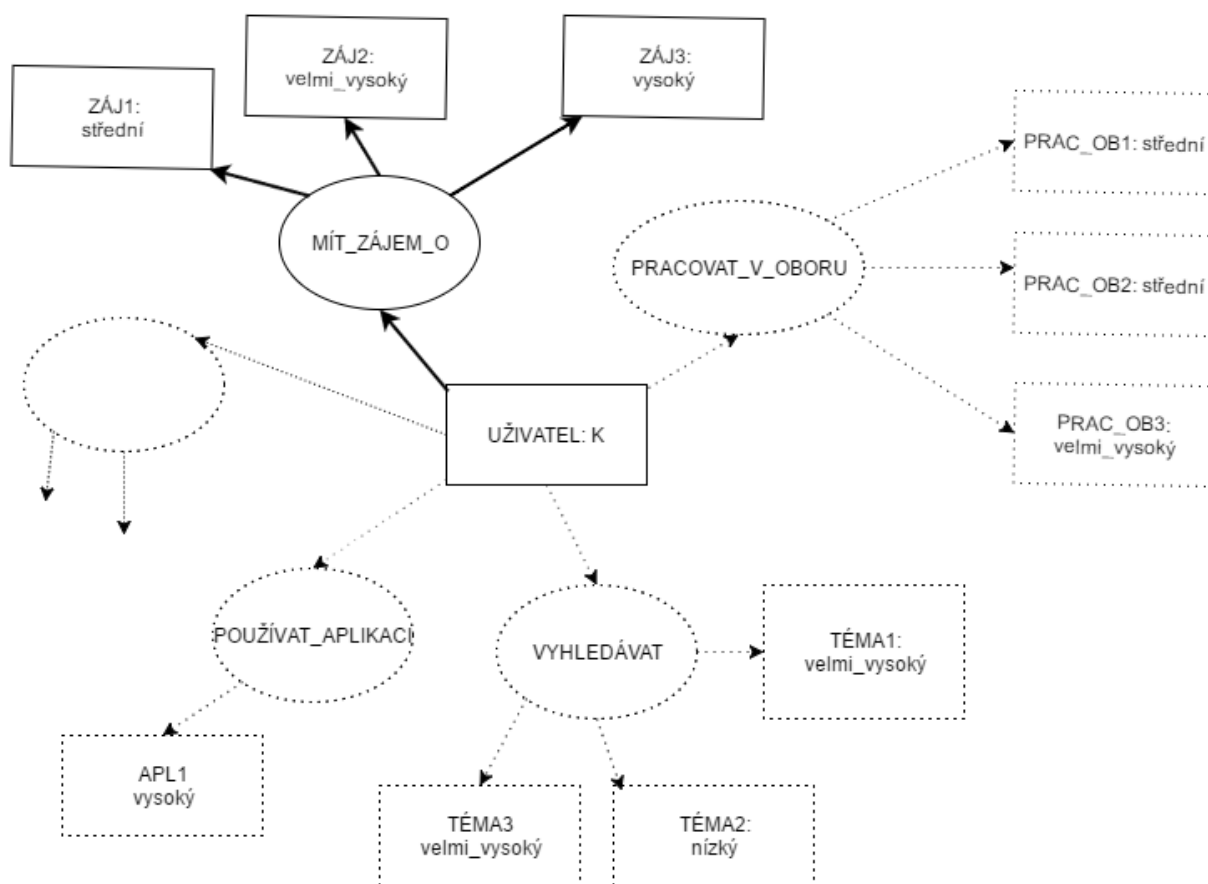
Vznikne konceptuální graf, ve kterém jsou uzly konceptů tvořeny klíčovými frázemi (VRML, C++, POČÍTAČOVÉ hry atp.) a referenty jsou fuzzy množiny reprezentující míru zájmu o dané téma (viz Obr. 4.9) ze strany uživatele. Uzly vztahů reprezentují zdroj informace obsažené v uzlu konceptu.

---

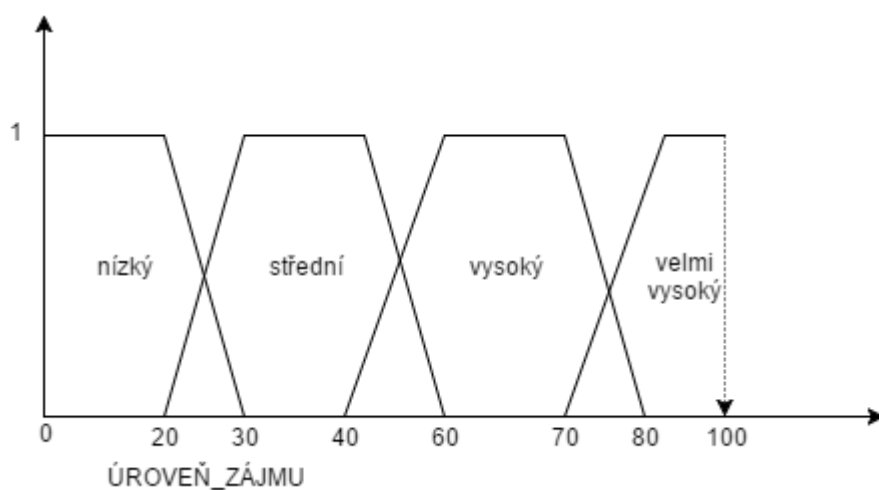
<sup>25</sup> Je třeba brát v potaz, že článek vyšel v roce 2000 a technologie se od té doby již posunuly vpřed.



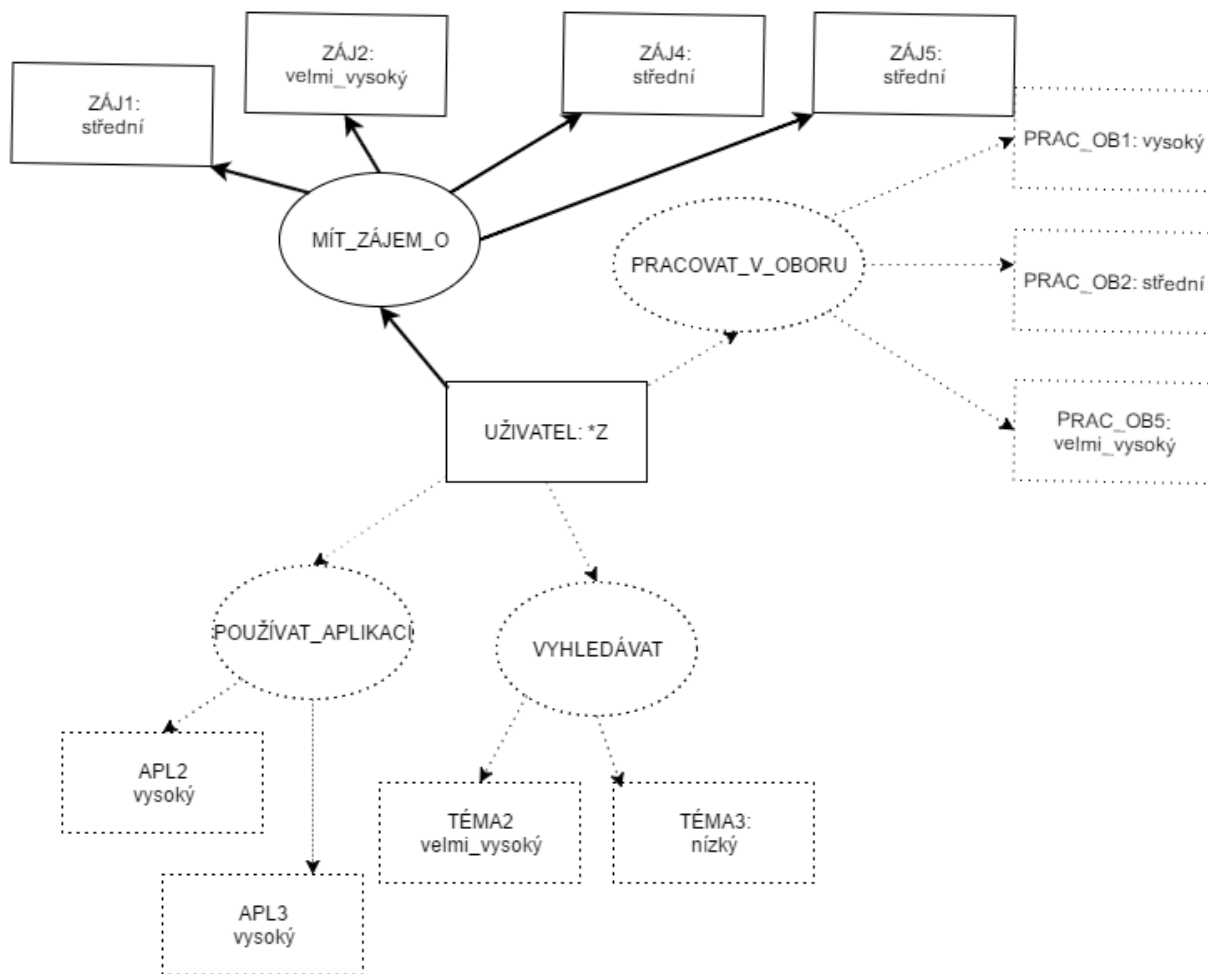
Obr. 4.7 Příklad konceptuálního grafu uživatele spadajícího do kategorie „Sdílené virtuální světy“ (Baldwin, Trevor a Tzanavari, 2000)



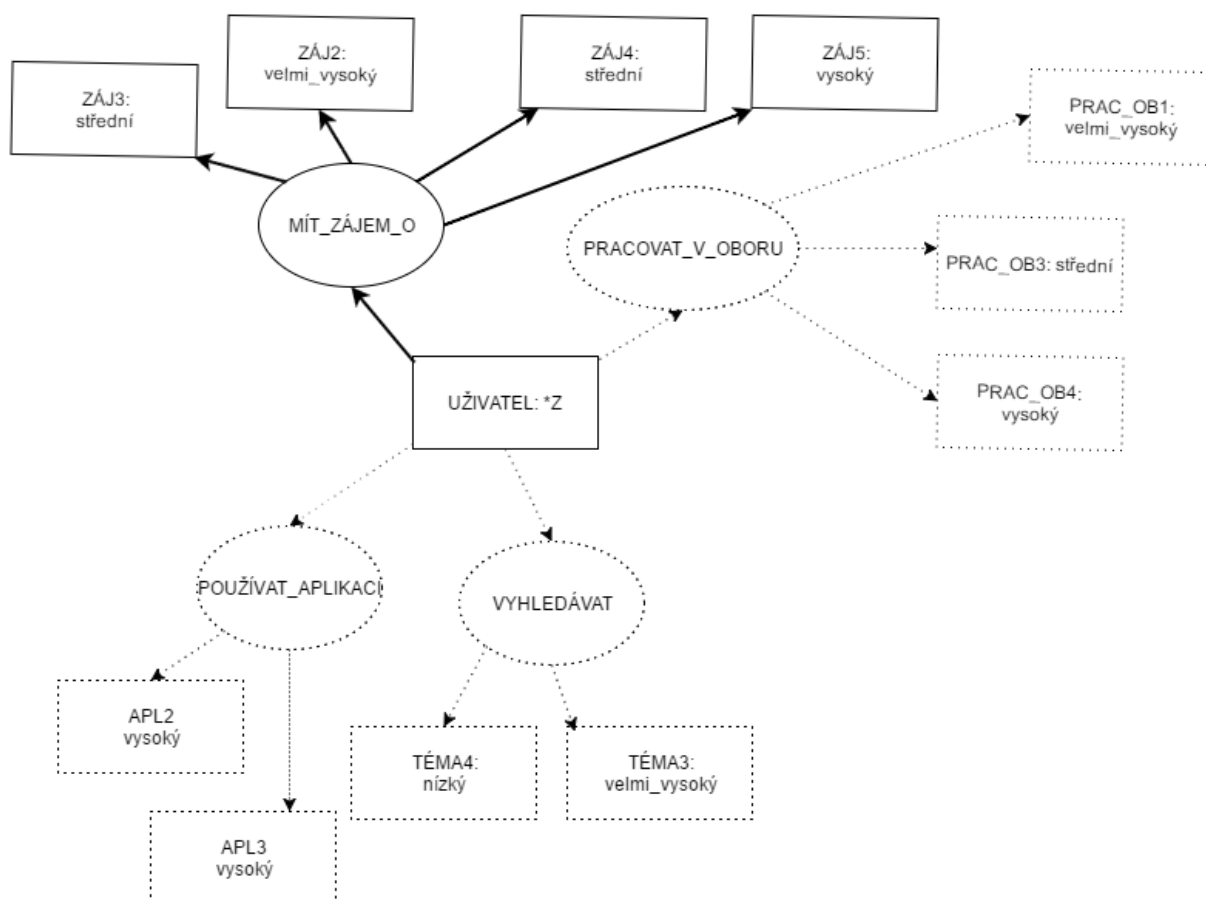
Obr. 4.8 Příklad abstrakčního konceptuálního grafu s uživatele k (Baldwin, Trevor a Tzanavari, 2000)



Obr. 4.9 Funkce příslušnosti definující fuzzy množinu obsaženou v referentu konceptu [ÚROVEŇ\_ZÁJMU] (Baldwin, Trevor a Tzanavari, 2000)



Obr. 4.10 Konceptuální graf prototypu P1: uzly konceptů a vztahů znázorněné přerušovanou čarou jsou odtrženy při hledání odpovědi na dotaz znázorněný v Obr. 1.10 (Baldwin, Trevor a Tzanavari, 2000)



Obr. 4.11 Konceptuální graf prototypu P2: uzly konceptů a relací vztahů znázorněné přerušovanou čarou jsou odtrženy při hledání odpovědi na dotaz znázorněný v Obr. 1.10 (Baldwin, Trevor a Zzanavari, 2000)

Porovnáním konceptuálního grafu s uživatele k (viz Obr. 4.8 ) se všemi konceptuálními grafy prototypů (P1 a P2 na Obr. 4.10 a Obr. 4.11) je získána informace o tom, do jaké kategorie a tedy i zóny aplikace Forum uživatel náleží. Dále je velmi užitečné odvodit informace o uživateli, které je pro systém neznámý, pomocí metody, jež byla popsána v kapitole Inferenční mechanika. Takový dotaz na informaci o uživateli je například dotaz „Jaký má uživatel k zájem o předmět zájmu č. 5?“, který je reprezentován konceptuálním grafem dotazu na Obr. 4.12. Pomocí inferenční mechaniky je možné získat odpověď: „střední“ pro prototyp P1 s podporou S1 a „vysoký“ pro prototyp P2 s podporou S2. Kombinací těchto výsledků je možno získat výsledek v rámci fuzzy množiny definující referenta konceptu [ÚROVEŇ\_ZÁJMU] a to buď v podobě přesné hodnoty, nebo slovního vyjádření například ve formě „poměrně vysoký“.



Obr. 4.12 Konceptuální graf Q reprezentující dotaz: „ Jaký má uživatel k zájem o předmět zájmu č. 5 ?“ (Baldwin, Trevor a Tzanavari, 2000)

#### 4.4 Teoretický příklad systému NaZahrádce

Cílem této kapitoly je na teoretické úrovni ilustrovat, jakým způsobem by bylo možné využít konceptuálních grafů při budování nového informačního systému. Systém NaZahrádce bude online dostupný komplexní systém určený pro osoby, které zajímá pěstování rostlin.

##### **Hlavní funkce systému NaZahrádce:**

- Platforma pro komunikaci a setkávání uživatelů
- Expertní systém
- Informační databáze

##### 4.4.1 Uživatelé

Systém bude zaměřen na laiky i odborníky. Uživatelé takového systému tvoří širokou a obtížně specifikovatelnou skupinu uživatelů internetu se zájmem o pěstování rostlin. Od uživatelů jsou očekávány odlišné zájmy (pěstování zeleniny, vzácných orchidejí, péče o ovocný sad) a také rozdílná úroveň expertízy v daném oboru. Je nutné předpokládat, že i tzv. laická veřejnost může mít v této oblasti velmi hluboké znalosti.

Bude nutné vytvořit model uživatelů. Prototypy mohou být vytvořeny na základě odhadů, a pokud se vytvořené prototypy neosvědčí, je možné díky modularitě konceptuálních grafů nastavení změnit. Znalosti, které bude potřeba o uživatelích získat, se budou týkat spíše jejich specifických zájmů o danou oblast, protože počítačová gramotnost pro takovýto druh systémů dle mého není zcela rozhodující. Bude vytvořen nějaký model hodnocení věrohodnosti uživatelů, který bude představován hodnotou, která se bude navyšovat v souvislosti s aktivitou v systému a s hodnocením od ostatních uživatelů.

Znalost o uživateli bude zanesena do konceptuálního grafu a na základě porovnávání grafů uživatelů bude možné spojovat uživatele s podobnými zájmy. Pokud by byly příspěvky, články a popřípadě témata diskuzí reprezentovány konceptuálními grafy, mohl by systém

nabízet uživateli příspěvky ostatních uživatelů ať již v podobě příspěvků v diskuzích nebo článků k jemu preferovaným tématům.

#### 4.4.2 **Expertní systém**

Cílem expertního systému bude poskytovat uživatelům rady ohledně pěstování rostlin. Systém by měl odpovědět na jakýkoliv dotaz, který může uživatel vznést: jakou péči která rostlina potřebuje, jaké ostatní rostliny snese ve svém okolí, jakou potřebuje půdu, kdy sázet, kdy sklízet a jak rozmnožovat? Nebo také jak si poradit se škůdci a nebo co může znamenat, když se rostlině rajčete i přes vhodnou závlivku krouží listy? Nebo pokud uživatel neví, jakým rostlinám by se nejlépe dařilo ve stinném, vlhkém koutu jeho zahrady?

#### **Báze znalostí**

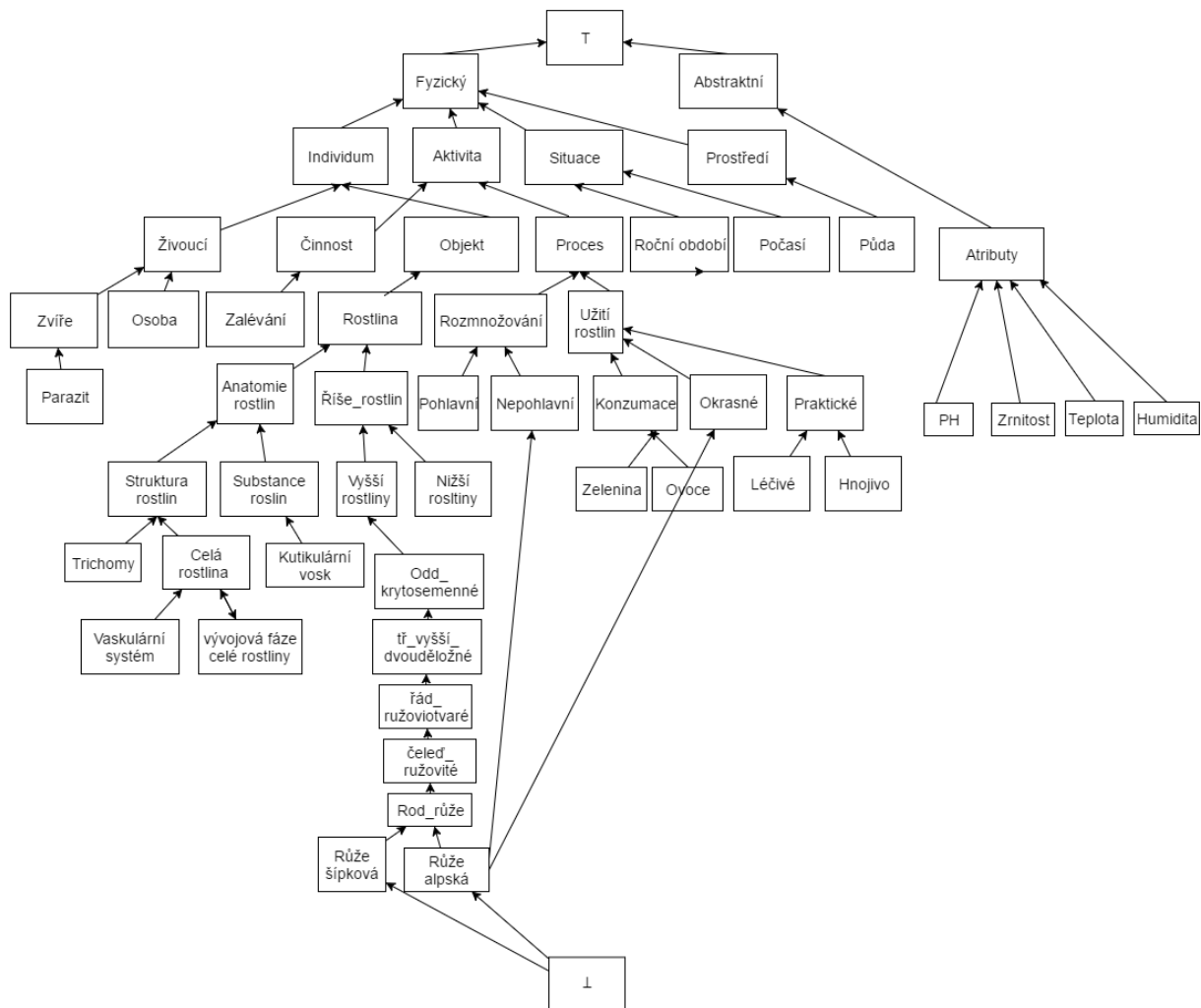
Prvním krokem bude vytvořit tzv. pomocný slovník pro budování expertního systému. Je tedy nutné získat svaz typů vztahů a svaz typů konceptů. Pro budování ontologie konceptů by bylo možno využít například volně dostupnou databázi Plant Ontology ([www.plantontology.org/](http://www.plantontology.org/)) nebo systém Agricultural Ontology Service Concept Server, což je ontologie obohacená o ontologické sémantické vztahy vycházející z původního tezauru AGROVOC. Celý systém je navíc v OWL, což znamená, že je volně mapovatelný do konceptuálních grafů. (Lauser, Sini, Liang, Keizer, Katz, 2006). Následně by bylo vhodné použít nějakou běžně užívanou taxonomii rostlin. Také by bylo možné vytěžit informační zdroje pro oblast pěstování rostlin a získat základní koncepty a konceptuální vztahy z jazykové analýzy textů. Zajímavou možností by bylo přidat do diskuze možnost ohodnotit přínos odpovědi na určitý dotaz, což je dneska již poměrně běžné. Odpovědi hodnocené jako kvalitní by bylo možné reprezentovat do podoby konceptuálního grafu a znalost implementovat do báze znalostí.

Je důležité, aby pomocný slovník obsahoval dostatečnou slovní zásobu pro pojmenování všech rostlin, procesů, které se s nimi mohou dít (uvadání, choroby, klíčení atd.), činnosti, které na nich provádějí lidé (sazení, roubování, křížení), dostatek atributů pro popsání lokalit a jejich vlastností (stinné stanoviště, jílovitá půda) a mnoho dalšího. Ilustrativní návrh svazů typů konceptů a vztahů je zobrazen na Obr. 4.13 a Obr. 4.14. Bylo by i možné vytvořit svazy typů funkcí a fuzzy typů. Pro fuzzy typy je nutné vytvořit funkce

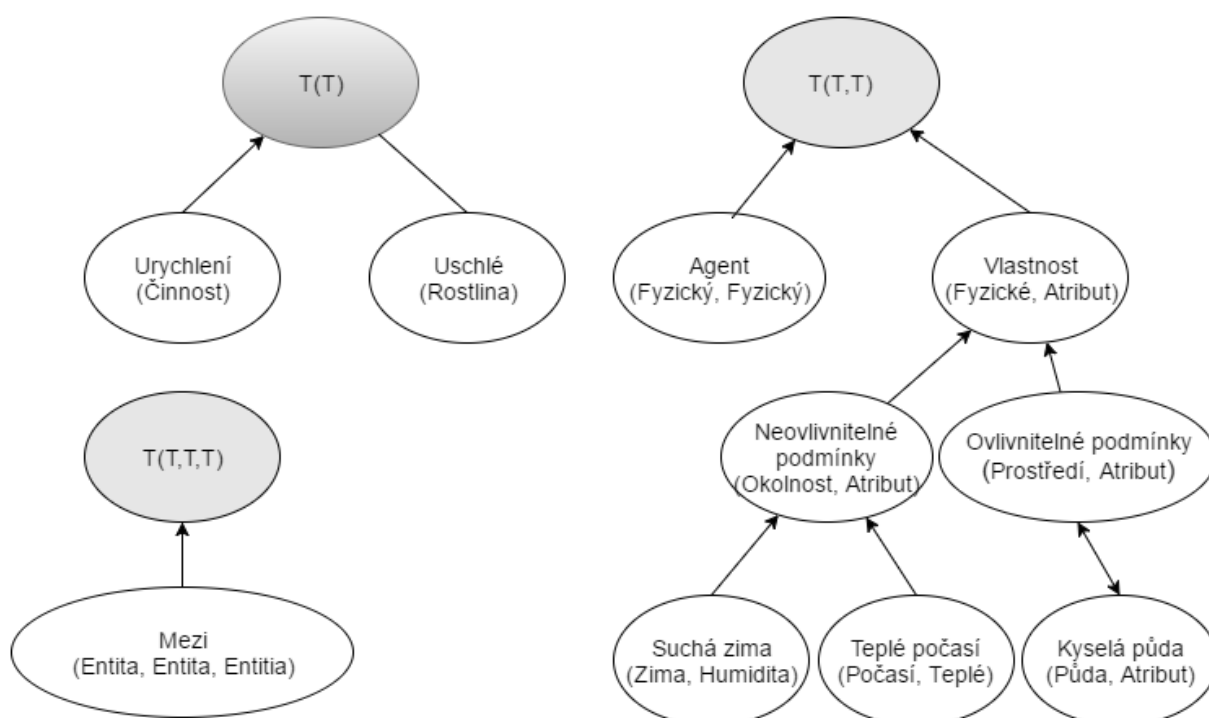


příslušnosti pro typy konceptů, které nelze určit přímo, ale mají jistou neurčitost. Takovým typem konceptu by mohla být například [HUMIDITA] (viz Obr. 4.15)

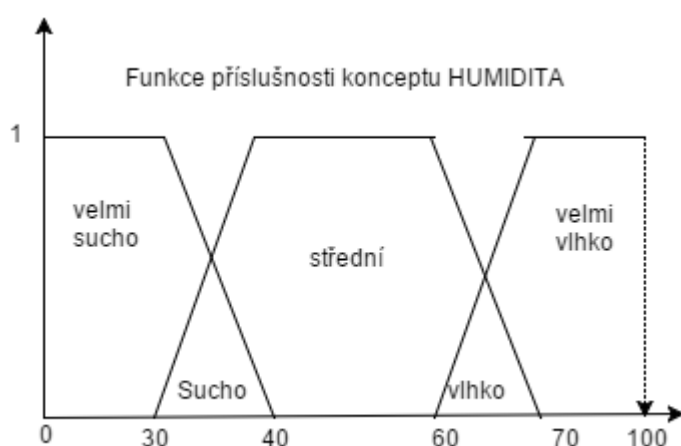
Následně bude třeba vytvořit systém, dle kterého budou vytvářeny kódy individuálních označovačů (například specifická jména uživatelů nebo obrázky uložené v bázi znalostí). Ve chvíli, kdy je hotový slovník, je nutné zkonstruovat vztahy, které definují vztah mezi typy a individuálními označovači.



Obr. 4.13 Ontologie konceptů systému NaZahrádce



Obr. 4.14 Ontologie vztahů systému NaZahrádce

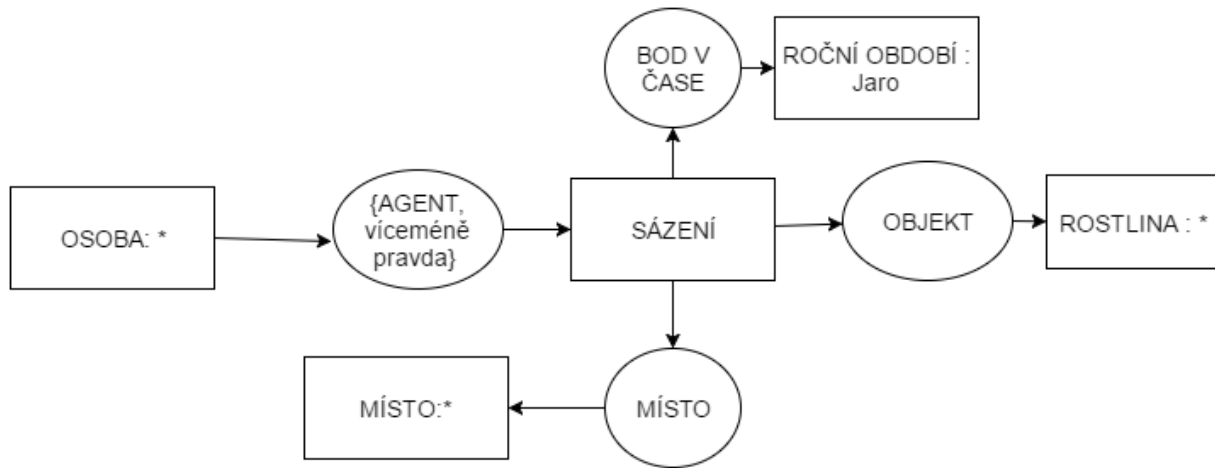


Obr. 4.15 Funkce příslušnosti konceptu HUMIDITA

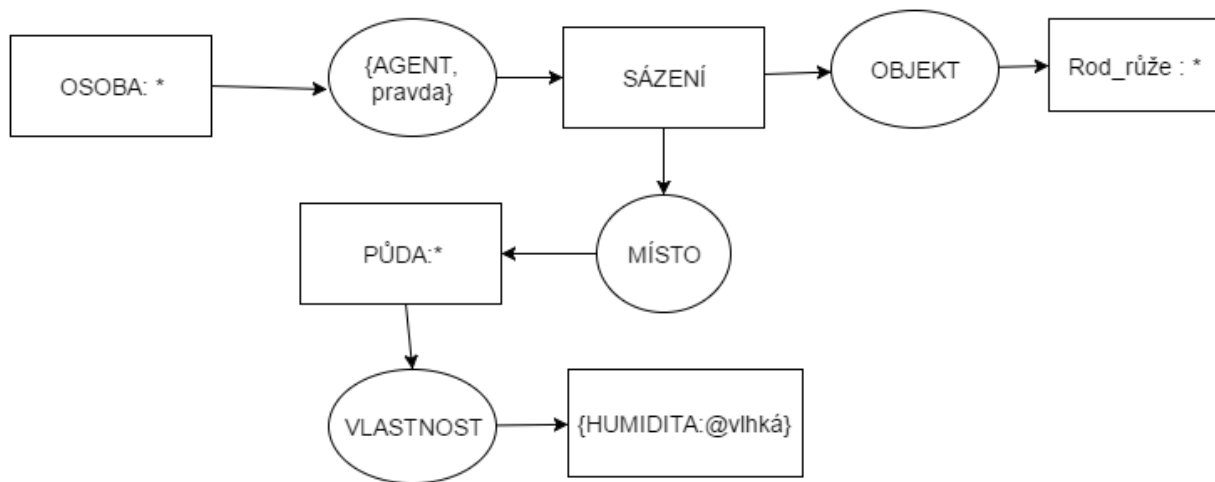
### Inferenční mechanismus

Inferenční mechanismus bude založen na operacích konceptuálních grafů. Navíc bude jednoduše vizualizovatelný, takže uživatelé budou mít možnost zjistit, co ovlivnilo výslednou odpověď. Uvedu příklad uživatelského dotazu: „Kdy se sázejí růže?“. Tento dotaz je převeden do formy konceptuálního grafu reprezentující dotaz Q (Obr. 4.20). Jsou vybrány dva grafy G1 a G2 reprezentující znalost z báze znalostí (viz Obr. 4.16 a Obr. 4.17), které mají jistou shodu s grafem Q. Následně je nejprve na grafu G1 a G2 provedena operace projekce a je získán graf Maximální společné projekce P (viz Obr. 4.18). z grafu P, G1 a G2 je operací *maximální*

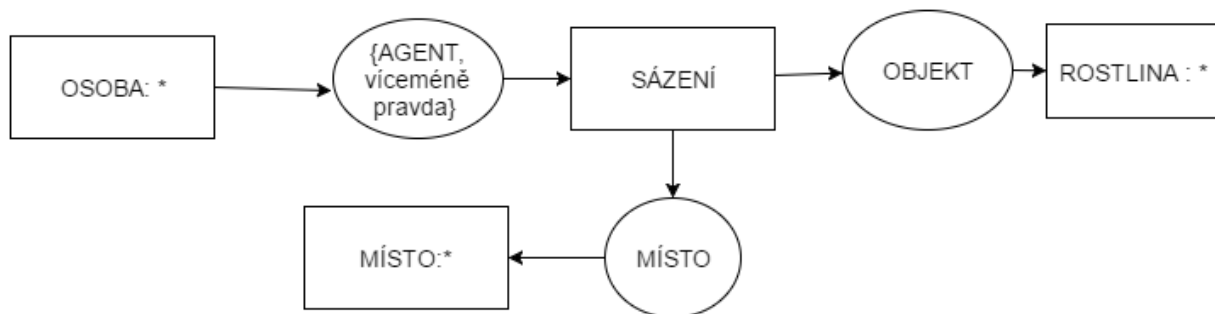
spojení získán graf M (Obr. 4.19). Následně bude použit inferenční mechanismus, kterak je popsán v kapitole 4.3.4) Tento popis je samozřejmě velmi zjednodušený, ale domnívám se, že dostatečně ilustruje, jak se dá ve znalostním systému odvozovat. Výsledná odpověď by zněla: „Je víceméně pravda, že růže se sází na jaře.“



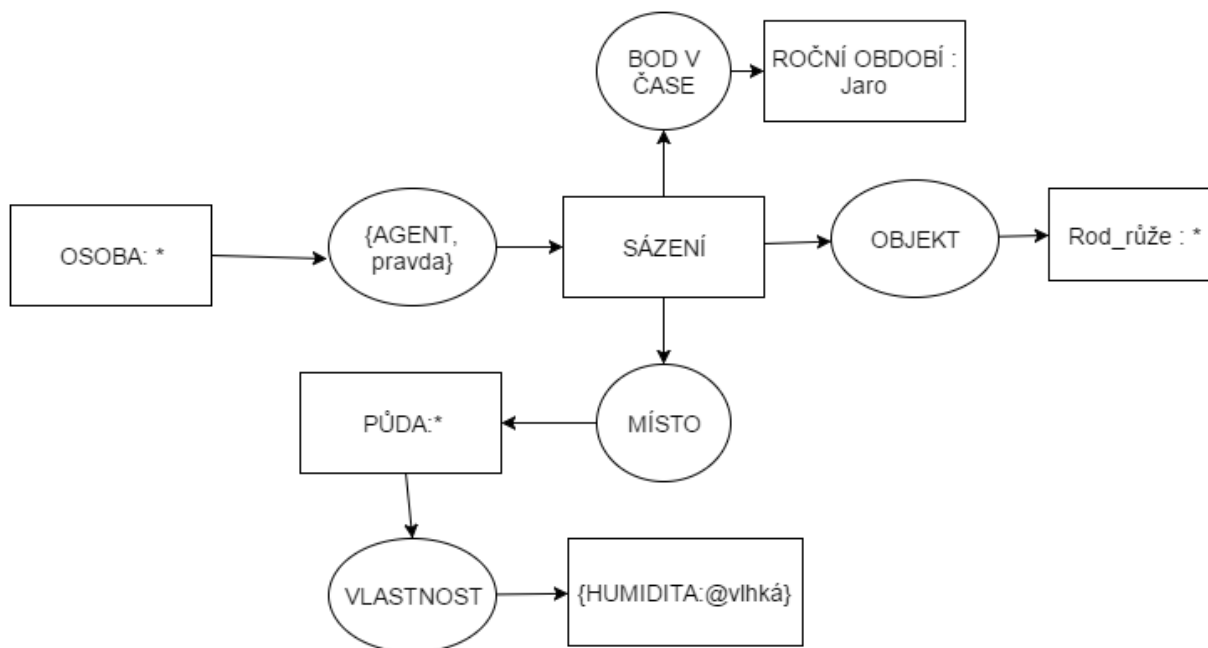
Obr. 4.16 Konceptuální graf reprezentující větu G1: "Rostliny se většinou sázejí na jaře."



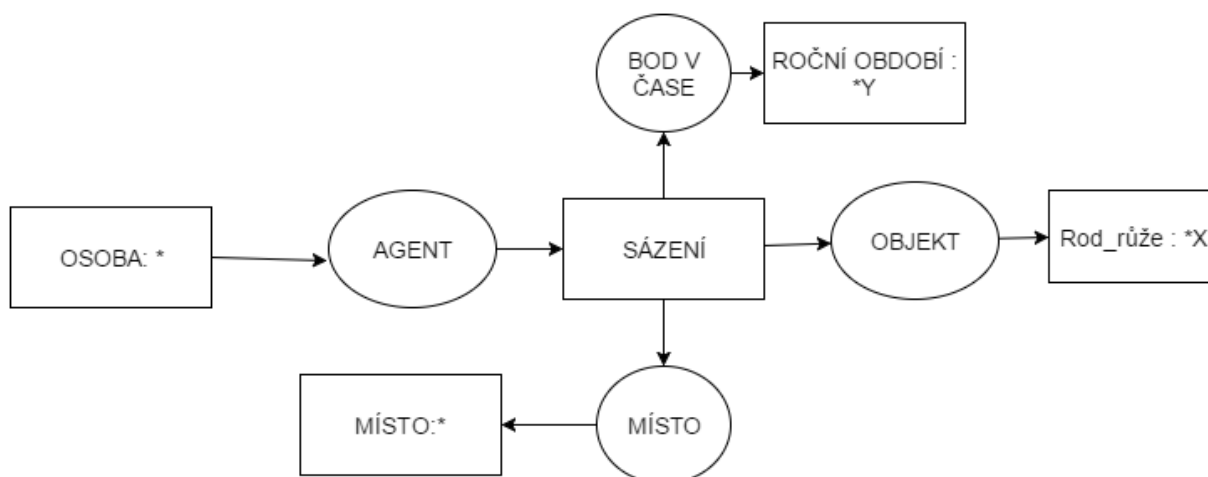
Obr. 4.17 Konceptuální graf reprezentující větu G2: "Růže se vždy sázejí do vlhké půdy."



Obr. 4.18 Maximální společná projekce P grafu G1 a G2



Obr. 4.19 Konceptuální graf M: Maximální spojení grafů G1 a G2 skrze jejich maximální společnou projekci P



Obr. 4.20 Konceptuální graf Q reprezentující dotaz: "V jakém ročním období se sázejí růže?"

#### 4.4.3 Informační databáze

Informační databáze by byla tvořena články z oblasti pěstitelství a zahrádkářství a to jak odborného tak laického charakteru. Bylo by možno reprezentovat jejich obsah ve formě jednoduchých konceptuálních grafů, což by bylo možno implementovat do inteligentního modulu vyhledávání informací. Informační databázi by bylo také možné vytěžovat při průběžné znalostní akvizici. Obrázky a videa by byly reprezentovány konceptuálními grafy vytvořenými na základě tagů uživatelů. Následně by bylo možné uživatelům nabízet relevantní zdroje k jejich dotazům nebo zájmům.

#### 4.4.4 Další funkce

V systému by mohly být implementovány normy, které by mohly například nepříhlášeným uživatelům nebo uživatelům s příliš nízkým hodnocením omezit možnosti přispívání do diskuzí nebo zveřejňovat články. Tyto normy lze zanést v podobě konceptuálních grafů. Systém by mohl být mapovatelný od standardizovaných jazyků reprezentace znalostí pro sémantický web. Navíc není problematické vybudovat na konceptuálních grafech multimediální systém. Jelikož je reprezentace znalostí pomocí konceptuálních grafů nezávislá na formátu a zakódování informace, je prostřednictvím individuálních označovačů poměrně jednoduché zakódovat zvuk, obrázek, video atd. Nebyl by problém rozšířit systém o nové moduly jako je například e-shop, který by nabízel uživatelům produkty dle jejich zájmů. Tento aspekt by mohl být využitelný i pro cílenou reklamu.

#### 4.4.5 Zhodnocení

Domnívám se, že je možné na základě formalismu konceptuálních grafů a jejich možnosti inferenčního mechanismu i mapování do jiných jazyků reprezentace znalostí vytvořit poměrně komplexní systém. Jako velmi cenné se mi jeví informace o uživatelích a bytí se mi přičí využití znalostí o uživatelích se záminkou jim poskytnout expertní radu a prostor pro setkávání pro obchodní účely, i tato varianta je možná.

## 5 Závěr

Konceptuální grafy jsou jazykem reprezentace znalostí, který se snaží vyjádřit význam formou, která má preciznost logiky, je člověkem čitelná i srozumitelná a zároveň aplikovatelná v počítačové vědě všeobecně. Díky tomuto základu jsou konceptuální grafy velmi vhodné pro zapracování do počítačových programů a v současné době se ve velké míře uplatňují v programech zaměřených na umělé odvozování.

Úvodní část práce představuje konceptuální grafy v kontextu dalších vědních oborů, které ovlivnily jejich vznik a vývoj. Je tak alespoň v základní podobě představeno celé spektrum teorií, které jsou v konceptuálních grafech syntetizovány. Podkapitola věnující se vývoji ilustruje, kterak byly postupně různé inovativní metody do teorie konceptuálních grafů přidávány, většinou v závislosti na vývoji oboru umělé inteligence v oblasti reprezentace znalostí. Podstatné je to, že od roku 1984, kdy Sowa zveřejnil své dílo *Conceptual Structures*, nebylo nutné zásadním způsobem do teorie ani formalismu konceptuálních grafů zasahovat. Veškeré inovace bylo možné implementovat jen s dílčími změnami. Druhá kapitola představuje formalismus konceptuálních grafů, tak jak je navrhl John F. Sowa. Jsou představeny formy notace, základní předpoklady validní konstrukce konceptuálních grafů, lambda kalkulus, koreferenční spojení a metody vyjádření kontextů pomocí zahrnutých grafů. Také je představeno pojetí ontologie dle Johna F. Sowy a význam ontologií pro informační vědu a reprezentaci znalostí a to speciálně v souvislosti s možnou budoucí podobou webu; tzv. sémantického webu.

Ontologie a konceptuální grafy dohromady tvoří mocný nástroj pro reprezentaci znalostí, ale v průběhu uplynulých dekád se objevilo několik nedokonalostí v možnostech reprezentovat znalosti pomocí konceptuálních grafů. Třetí kapitola této práce představuje implementaci fuzzy logiky pro práci s neurčitostí a pravděpodobností, funkci pro vyjádření funkčních závislostí, zajímavou reprezentaci času a notaci grafů CGIF standardizovanou ISO normou ISO/IEC 24707:2007 jako jeden z jazyků obecné logiky. Avšak především představuje kánonická pravidla a základní operace specializace a zobecnění, které společně s tzv. podporou, která je v této kapitole také popsána, tvoří základ pro budování expertních systémů.

Poslední kapitola přináší přehled možné implementace konceptuálních grafů v praxi. Zevrubně vysvětluje přínosy využití v oblastech expertních systémů, akvizice znalostí, zpracování textů, zpracování přirozených jazyků, managementu znalostí atd. Zvláštní podkapitoly se věnují sémantickému webu a také využití konceptuálních grafů nejen jako jazyka komunikace mezi počítači a lidmi, ale také mezi počítači a systémy a především i mezi lidmi z různých oborů, kdy může jeho jednoduchá forma diagramu leckdy velmi dobře přispět k vzájemnému porozumění. Obsáhlejší podkapitola se věnuje užití konceptuálních znalostí pro modelování uživatelů. Na tomto jednoduchém příkladu z praxe je ukázáno, kterak může fungovat mechanismus odvozování znalostí. Kapitulu uzavírá popis teoretického

systému NaZahrádce, na kterém je přehledným a jednoduchým způsobem sumarizováno, jak je možno konceptuálních grafů využít při návrhu informačního a expertního systému.

Zajímavé je, že jsem při podrobné analýze zdrojů pojednávajících o konceptuálních grafech nenalezla žádné zásadní nevýhody jejich užití. Pokud se v nějaké implementaci objevily nedostatky, jež bránily například kvalitní reprezentaci, bylo možné tyto nedostatky odstranit pomocí nějakého rozšíření nebo inovace, která však nijak zásadním způsobem nezměnila formalismus nebo teoretický základ konceptuálních grafů. Jediné co se mi jeví trochu problematické je, že model klade přílišný důraz na syntax, kdy věty nesoucí stejný význam jsou většinou vyjádřené jiným grafem. Ovšem pro reprezentaci znalostí tento fakt dle mého není zcela zásadní.

Mezi výhody jistě patří to, že jsou konceptuální grafy mapovatelné do logiky prvního řádu, přeložitelné do jiných formálních jazyků reprezentace znalostí a především i jednoduše vizualizovatelné bez nutnosti je komplikovaně převádět na grafické zobrazení. Velikou výhodou je také jejich univerzálnost, kdy jich může být jako systému logiky využito pro popis nebo definici čehokoliv, co je implementováno v digitálním počítači. Mezi jejich hlavní výhody oproti některým dalším formalismům reprezentace znalostí patří jejich flexibilita, modularita a schopnost reprezentovat různé úrovně znalostí.

Dle mého názoru jsou konceptuální grafy výborným nástrojem reprezentace znalostí nejrůznějšího druhu, modelování i odvozování znalostí. Mohou pomoci specifikovat modely a architekturu, model jejich uživatelů, samotné znalosti, procesy, které nad informacemi probíhají i popsat okolí informačního systému a mnoho dalšího. Mapovatelnost konceptuálních grafů do standardizovaných jazyků sémantického webu a jejich základ v teorii ontologií společně s tím, že elegantním způsobem kombinují heuristické i přísně logicko-formální přístup ke znalostem dle mého názoru dává konceptuálním grafům velké šance pro jejich budoucí využití.

## Seznam použité literatury

1. ALLEN, Robert B, 1990. User models: theory, method, and practice. *International Journal of Man-Machine Studies*. Volume 32, Issue 5 , s. 511 - 543. ISSN 0020-7373.
2. BALDWIN, J. F., 2005. A theory of mass assignments for artificial intelligence. In: *Fuzzy Logic and Fuzzy Control: IJCAI '91 Workshops on Fuzzy Logic and Fuzzy Control Sydney, Australia, August 24, 1991 Proceedings* [online]. 1. Berlin: Springer, s. 22-34 [cit. 2016-07-31]. DOI: 10.1007/3-540-58279-7\_16. ISBN 978-3-540-48602-2. ISSN 0302-9743. Dostupné z: [http://link.springer.com/10.1007/3-540-58279-7\\_16](http://link.springer.com/10.1007/3-540-58279-7_16)
3. BALDWIN, J.F., 1993. Fuzzy, probabilistic and evidential reasoning in Fril. In: *Fuzzy Logic and Fuzzy Control: IJCAI '91 Workshops on Fuzzy Logic and Fuzzy Control Sydney, Australia, August 24, 1991 Proceedings* [online]. 1. Berlín: IEEE, s. 459-464 [cit. 2016-07-31]. DOI: 10.1109/FUZZY.1993.327511. ISBN 0-7803-0614-7. ISSN 0302-9743. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=327511>
4. BALDWIN, James F., Trevor P. MARTIN a Aimilia TZANAVARI, 2000. User Modelling Using Conceptual Graphs for Intelligent Agents. In: *Conceptual Structures: Logical, Linguistic, and Computational Issues: 8th International Conference on Conceptual Structures, ICCS 2000, Darmstadt, Germany, August 14-18, 2000. Proceedings*. Berlin: Springer, s. 193. DOI: 10.1007/10722280\_13. ISBN 978-3-540-44663-7. ISSN 0302-9743. Dostupné také z: [http://link.springer.com/10.1007/10722280\\_13](http://link.springer.com/10.1007/10722280_13)
5. BARTL, David, 2006. *Úvod do teorie množin a logiky 2* [online]. Verze ke dni 10. 12. 2008. Ostrava: Ostravská univerzita v Ostravě [cit. 2016-07-30]. Dostupné z: <http://www1.osu.cz/~bartl/pubs/?index=0085&fullt=10>
6. BARWISE, Jerry a Jon PERRY, 1986. Situations and Attitudes. *The Philosophical Quarterly*. MIT Press, 36(143), 292-. DOI: 10.2307/2219775. ISSN 00318094. Dostupné také z: <http://pq.oxfordjournals.org/cgi/doi/10.2307/2219775>
7. BEIERLE, C., U. HEDTSTÜCK, U. PLETAT, P.H. SCHMITT a J. SIEKMANN, 1992. An order-sorted logic for knowledge representation systems. *Artificial Intelligence* [online]. 55(2-3), 149-191 [cit. 2016-07-30]. DOI: 10.1016/0004-3702(92)90055-3. ISSN 00043702. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/0004370292900553>
8. BĚLOHLÁVEK, Radim a Vilém VYCHODIL, 2005. *What is a fuzzy concept lattice?*. Olomouc: Department of Computer Science, Palacky University. Dostupné také z: [http://belohlavek.inf.upol.cz/publications/BeVy\\_Wifcl.pdf](http://belohlavek.inf.upol.cz/publications/BeVy_Wifcl.pdf)
9. BERNERS-LEE, Tim 2001. Conceptual Graphs and the Semantic Web. *Reflections on Web Architecture* [online]. [cit. 2016-07-31]. Dostupné z: <https://www.w3.org/DesignIssues/CG.htm>
10. BERNERS-LEE, Tim, James HENDLER a Ora LASSILA, 2001. The Semantic Web: A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American* [online]. [cit. 2016-07-31]. Dostupné z: [https://www-sop.inria.fr/acacia/cours/essi2006/Scientific%20American\\_%20Feature%20Article\\_%20The%20Semantic%20Web\\_%20May%202001.pdf](https://www-sop.inria.fr/acacia/cours/essi2006/Scientific%20American_%20Feature%20Article_%20The%20Semantic%20Web_%20May%202001.pdf)
11. BORST, Willem Nico, 1997. *Construction of Engineering Ontologies for Knowledge Sharing and Reuse* [online]. Enschede: W. N. Borst [cit. 2016-07-30]. ISBN 90-365-0988-2. Dostupné z: <http://eprints.eemcs.utwente.nl/17377/01/t0000004.pdf>
12. BOUZEK, Jan a Zdeněk KRATOCHVÍL, 1994. *Od mýtu k logu*. V Praze: Herrmann & synové.



13. CAO, Tru H. a Peter N. CREASY, 2000. Fuzzy types: a framework for handling uncertainty about types of objects. *International Journal of Approximate Reasoning* [online]. 2000, 25(3), 217-253 [cit. 2016-07-31]. DOI: 10.1016/S0888-613X(00)00055-4. ISSN 0888613x. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0888613X00000554>
14. CAO, Tru H., Peter N. CREASY a Vilas WUWONGSE, 1997. Fuzzy unification and resolution proof procedure for fuzzy conceptual graph programs. In: *Conceptual Structures: Fulfilling Peirce's Dream: Fifth International Conference on Conceptual Structures, ICCS'97 Seattle, Washington, USA, August 3–8, 1997 Proceedings*. Berlin: Springer, s. 386-400. DOI: 10.1007/BFb0027885. ISBN 978-3-540-69424-3. ISSN 0302-9743. Dostupné také z: <http://link.springer.com/10.1007/BFb0027885>
15. CAO, Tru Hoang, 2010. *Conceptual graphs and fuzzy logic: a fusion for representing and reasoning with linguistic information* [online]. 1. Berlin: Springer-Verlag [cit. 2016-07-30]. ISBN 978-364-2140-877. Dostupné z: <http://www.springer.com/us/book/9783642140860>
16. CECCATO, Silvio, 1961. *Linguistic Analysis and Programming for Mechanical Translation, Gordon and Breach*. New York: Gordon and Breach.
17. COX, Lisa, Harry S. DELUGACH a David SKIPPER, 2001. Dependency Analysis Using Conceptual Graphs. In: *In Proceedings of the 9th International Conference on Conceptual Structures, ICCS 2001, Conceptual Structures: Broadening the Base, July 30 - August 3, Stanford University, California, USA*. Stanford University: Cali, 2001, s. 14. Dostupné také z: <http://ceur-ws.org/Vol-41/Cox.pdf>
18. CROITORU, Madalina, Nir OREN, Simon MILES a Michael LUCK, 2012. Graphical norms via conceptual graphs. *Knowledge-Based Systems* [online]. 29, 31-43 [cit. 2016-07-31]. DOI: 10.1016/j.knosys.2011.06.025. ISSN 09507051. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0950705111001432>
19. ČADA, Roman, Tomáš KAISER a Zdeněk RYJÁČEK, 2004. *Diskrétní matematika*. Plzeň: Západočeská univerzita v Plzni. ISBN 80-708-2939-7. Dostupné také z: <http://www.cam.zcu.cz/~ryjacek/students/DMA/skripta/dma.pdf>
20. ČERNÍK, Václav, 2000. *K pojmu bytia: Problém rekonštrukcie klasickej ontológie*. Bratislava: Honner, 150 s. ISBN 8096839918.
21. ČERVENÝ, Vlastimil, 1999. Vyhledávání v databázích plných textů. *Národní knihovna Knihovnicka revue*. Praha: Národní knihovna, 10(1), 6-12. ISSN 1214-0678. Dostupné také z: <http://full.nkp.cz/nkk/Nkk9901/9901006.html>
22. DIENG-KUNTZ, Rose a Olivier CORBY, 2005. Conceptual Graphs for Semantic Web Applications. In: *Proceeding ICCS'05 Proceedings of the 13th international conference on Conceptual Structures: common Semantics for Sharing Knowledge*. Berlin: Springer, s. 19-50. DOI: 10.1007/11524564\_2. ISBN 3-540-27783-8. Dostupné také z: [http://link.springer.com/10.1007/11524564\\_2](http://link.springer.com/10.1007/11524564_2)
23. DOBREV, P. a A. STRUPCHANSKA, 2005. Conceptual Graphs and Annotated Semantic Web Pages. In: *Conceptual Graphs and Annotated Semantic Web Pages*. s. 54-65. Dostupné také z: <http://lml.bas.bg/~albena/publications/iccs05.pdf>
24. DUŽÍ, Marie, 2012. *Logika pro informatiky (a příbuzné obory): učební text* [online]. Ostrava: VŠB-TU Ostrava [cit. 2016-07-30]. ISBN 978-80-248-2662-2.
25. DVOŘÁK, Jiří, 2004. *Expertní systémy*. Brno: Vysoké učení technické v Brně. Dostupné také z: <http://www.uai.fme.vutbr.cz/~jdvorak/Opory/ExpertniSystemy.pdf>
26. FERNÁNDEZ, Milagros, Eric de la CLERGERIE a Manuel VILARES, 2008. Mining conceptual graphs for knowledge acquisition. In: *Proceeding of the 2nd ACM workshop on Improving non english web searching - iNEWS '08*. New York: ACM Press, s. 25-32. DOI: 10.1145/1460027.1460032. ISBN 9781605584164. Dostupné také z: <http://portal.acm.org/citation.cfm?doid=1460027.1460032>
27. FISHER, Gerhard, 2001. User Modeling in Human–Computer Interaction. *User Modeling and User-Adapted Interaction*. vol. 11, Issue 1-2, s. 65 - 86. ISSN 0924-1868.

28. FOGUEM, Bernard Kamsu, Germaine Tchuente FOGUEM a Clovis FOGUEM, 2014. Using conceptual graphs for clinical guidelines representation and knowledge visualization. *Information Systems Frontiers* [online]. Hingham: Kluwer Academic Publishers, 16(4), 571-589 [cit. 2016-07-31]. DOI: 10.1007/s10796-012-9360-2. ISSN 1387-3326. Dostupné z: <http://link.springer.com/10.1007/s10796-012-9360-2>
29. FOLTÝNEK, Tomáš a Jana DANNHOFFEROVÁ, 2011. *Teorie grafů*. Brno: Mendelova univerzita v Brně. ISBN 978-80-7375-500-3.
30. GANTER, Bernhard, Gerd STUMME a Rudolf WILLE, 2005. *Formal Concept Analysis: Foundations and Applications*. 2005. Berlín: Springer-Verlag. ISBN 978-3-540-31881-1. 10.1007/978-3-540-31881-1.
31. GANTER, Bernhard, Gerd STUMME a Rudolf WILLE, 2006. *Formal concept analysis: foundations and applications*. Berlin: Springer. Lecture notes in computer science. ISBN 3540278915.
32. GENEST, David a Michel CHEIN, 2005. A content-search information retrieval process based on conceptual graphs. *Knowledge and Information Systems* [online]. 8(3), 292-309 [cit. 2016-07-30]. DOI: 10.1007/s10115-004-0179-0. ISSN 0219-1377. Dostupné z: <http://link.springer.com/10.1007/s10115-004-0179-0>
33. GÓMEZ, Manuel Montes, Aurelio López LÓPEZ a Alexander GELBUKH, 2001. Information Retrieval with Conceptual Graph Matching. In: *Database and Expert Systems Applications: 11th International Conference, DEXA 2000 London, UK, September 4–8, 2000 Proceedings*. Berlin: Springer, 312 -321. DOI: 10.1007/3-540-44469-6\_29. ISBN 978-3-540-44469-5. Dostupné také z: [http://link.springer.com/10.1007/3-540-44469-6\\_29](http://link.springer.com/10.1007/3-540-44469-6_29)
34. GRUBER, Thomas R., 1993. A translation approach to portable ontology specifications. *Knowledge Acquisition*. 5(2), 199-220. DOI: 10.1006/knac.1993.1008. ISSN 10428143. Dostupné také z: <http://linkinghub.elsevier.com/retrieve/pii/S1042814383710083>
35. HAJÍČOVÁ, Eva, 2006. K některým otázkám závislostní gramatiky. *Slovo a slovesnost*. Praha: Ústav pro jazyk český Akademie věd České republiky, 67(1), 3-26. ISSN 0037-7031. Dostupné také z: <http://dlib.lib.cas.cz/>
36. HAYS, David G, 1964. Dependency theory: a formalism and some observations. *Language*. 40(4), 511-525.
37. HEMA, N. a S. JUSTUS, 2015. Conceptual Graph Representation Framework for ICD-10. *Procedia Computer Science* [online]. Elsevier, 50, 635-642 [cit. 2016-07-31]. DOI: 10.1016/j.procs.2015.04.097. ISSN 18770509. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S1877050915005980>
38. HENSMAN, Svetlana, 2005. *Constructing conceptual graphs using linguistic resources* [online]. Dublin: University College Dublin [cit. 2016-07-31].
39. HENSMAN, Svetlana, 2004a. *Automatically building conceptual graphs using VerbNet and WordNet* [online]. Dublin: University College Dublin, [cit. 2016-07-31].
40. HENSMAN, Svetlana, 2004b. *Construction of conceptual graph representation of texts* [online]. Dublin: University College Dublin, [cit. 2016-07-31].
41. HOLUB, Josef a Stanislav LYER, 1992. *Stručný etymologický slovník jazyka českého: se zvláštním zřetelem k slovům kulturním a cizím*. 4. vyd. Praha: Státní pedagogické nakladatelství. Odborné slovníky. ISBN 80-042-3715-0.
42. CHEIN, Michel a Marie-Laure MUGNIER, 2009. *Graph based knowledge representation: Computational Foundations of Conceptual Graphs*. London: Springer. ISBN 978-184-8002-869. DOI 10.1007/978-1-84800-286-9.

43. CHEIN, Michel a Marie-Laure MUGNIER, 1992. Conceptual Graphs: fundamental notions. *Revue d'Intelligence Artificielle* [online]. Cachan: Lavoisier, 6(4), pp 365--406. [cit. 2016-07-30]. ISSN 1958-5748. Dostupné z: [http://www.info2.uqam.ca/~lounis\\_h/dic9250/Supplement\\_GConceptuels.pdf](http://www.info2.uqam.ca/~lounis_h/dic9250/Supplement_GConceptuels.pdf)
44. CHEIN, Michel a Marie-Laure MUGNIER, 2014. *Conceptual Graphs Are Also Graphs*. , 1. DOI: 10.1007/978-3-319-08389-6\_1. Dostupné také z: [http://link.springer.com/10.1007/978-3-319-08389-6\\_1](http://link.springer.com/10.1007/978-3-319-08389-6_1)
45. CHEIN, Michel, Marie-Laure MUGNIER a Madalina CROITORU, 2013. Visual reasoning with graph-based mechanisms: the good, the better and the best. *The Knowledge Engineering Review* [online]. 28(03), 249-271 [cit. 2016-07-31]. DOI: 10.1017/S0269888913000234. ISSN 0269-8889. Dostupné z: [http://www.journals.cambridge.org/abstract\\_S0269888913000234](http://www.journals.cambridge.org/abstract_S0269888913000234)
46. CHOMSKY, Noam, 1975. *Syntactic structures*. 11. Paris: Mouton. ISBN 90-279-3385-5. Dostupné také z: [http://ewan.website/egg-course-1/readings/syntactic\\_structures.pdf](http://ewan.website/egg-course-1/readings/syntactic_structures.pdf)
47. CHURCH, Alonzo, 1941. The calculi of lambda-conversion. *Annals of Mathematics studies*. Princeton: Princeton University Press, no. 6, 77. DOI: 10.2307/2267126. ISSN 0022-4812. Dostupné také z: [http://www.journals.cambridge.org/abstract\\_S0022481200106462](http://www.journals.cambridge.org/abstract_S0022481200106462)
48. ISO/IEC 24707, 2007. *Information technology — Common Logic (CL): a framework for a family of logic-based languages*. Geneva: ISO copyright office, 2007. Dostupné také z: <http://standards.iso.org/ittf/licence.html>
49. JIMÉNEZ, Sabino Miranda, Alexander GELBUKH a Grigori SIDOROV, 2013. Summarizing Conceptual Graphs for Automatic Summarization Task. In: *Conceptual Structures for STEM Research and Education: 20th International Conference on Conceptual Structures, ICCS 2013, Mumbai, India, January 10-12, 2013. Proceedings*. Berlin: Springer, 245 -253. DOI: 10.1007/978-3-642-35786-2\_18. ISBN 978-3-642-35786-2 Series Title. ISSN 0302-9743. Dostupné také z: [http://link.springer.com/10.1007/978-3-642-35786-2\\_18](http://link.springer.com/10.1007/978-3-642-35786-2_18)
50. KADLECOVÁ, Anna, PROKOPOVÁ, Kateřina (ed.), 2014. Komponentová analýza významu. In: *Encyklopedie lingvistiky* [online]. Olomouc: Univerzita Palackého v Olomouci, 2014 [cit. 2016-07-30]. Dostupné z: [http://oltk.upol.cz/encyklopedie/index.php5/Komponentová1\\_analýza\\_významu](http://oltk.upol.cz/encyklopedie/index.php5/Komponentová1_analýza_významu)
51. KAMSU-FOGUEM, Bernard, Gayo DIALLO a Clovis FOGUEM, 2013. Conceptual graph-based knowledge representation for supporting reasoning in African traditional medicine. *Engineering Applications of Artificial Intelligence* [online]. 26(4), 1348-1365 [cit. 2016-07-31]. DOI: 10.1016/j.engappai.2012.12.004. ISSN 09521976. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0952197612003120>
52. KENT, Allen a James G. WILLIAMS, 1996. *Encyclopedia of Computer Science and Technology: Volume 34 - Supplement 19: Artificial Intelligence in Education to An Undergraduate Course? Advising Expert System in Industrial Engineering*. New York: CRC Press. ISBN 978-082-4722-876.
53. KIRIKOVA, Marite, Wita WOJTKOWSKI a Janis GRUNDSPENKIS (eds.), 2003. *Information Systems Development: Advances in Methodologies, Components, and Management* [online]. 1. Boston: Springer US, 459 s. [cit. 2016-07-31]. ISBN 978-146-1501-671. ISSN DOI 10.1007/978-1-4615-0167-1.
54. KOGNITIVNÍ SERVER [online], 2010. Hradec Králové : Univerzita Hradec Králové. [cit. 2010-08-04]. Výkladový slovník. Dostupné z WWW: <<http://fim.uhk.cz/cogn/>>.
55. KOLODNER, Janet L, 2002. The "Neat" and the "Scruffy" in Promoting Learning From Analogy: We Need to Pay Attention to Both. In: *Journal of the Learning Sciences*. 11(1), s. 139-152. DOI: 10.1207/S15327809JLS1101\_7. ISSN 1050-8406. Dostupné také z: [http://www.tandfonline.com/doi/abs/10.1207/S15327809JLS1101\\_7](http://www.tandfonline.com/doi/abs/10.1207/S15327809JLS1101_7)
56. KUČERA, Radan, 2010. *Základy teorie svazů*. 30 s. Dostupné také z: <http://www.math.muni.cz/~kucera/texty/Svazy2010.pdf>

57. KUČEROVÁ, Helena, 2003. Expertní systém. In: *KTD: Česká terminologická databáze knihovnictví a informační vědy (TDKIV)* [online]. Praha : Národní knihovna ČR, 2003- [cit. 2016-07-31]. Dostupné z: [http://aleph.nkp.cz/F/?func=direct&doc\\_number=000000099&local\\_base=KTD](http://aleph.nkp.cz/F/?func=direct&doc_number=000000099&local_base=KTD).
58. KUCHYŇKA, Petr a Jiří RACLAVSKÝ, 2014. *Pojmy a vědecké teorie*. Brno: Masarykova univerzita. ISBN 978-80-210-6791-2.
59. LIU, P. Q. a Y. TIAN, 2013. Study on Knowledge Representation of Temporal Conceptual Graph in the Context". *Applied Mechanics and Materials*. Trans Tech Publications, 411-414, 251-254. DOI: 10.4028/www.scientific.net/AMM.411-414.272. ISSN 1662-7482. Dostupné také z: <http://www.scientific.net/AMM.411-414.251>
60. LUKASOVÁ, Alena, 2010. *Formální reprezentace znalostí*. V Ostravě: Ostravská univerzita. Universum (Ostravská univerzita). ISBN 978-80-7368-900-1.
61. MAJUMDAR, Arun k., SOWA, John F. (ed.),2003. Analogical Reasoning. MOOR, Aldo de. *Conceptual Structures for Knowledge Creation and Communication: 11th International Conference on Conceptual Structures, ICCS 2003, Dresden, Germany, July 21-25, 2003, Proceedings* [online]. Berlín: Springer-Verlag, s. 16-36 [cit. 2016-07-30]. DOI: 10.1007/b11835. ISBN 978-3-540-45091-7. Dostupné z: <http://www.jfsowa.com/pubs/analog.htm>
62. MAŘÍK, Vladimír, Olga ŠTĚPÁNKOVÁ a Jiří LAŽANSKÝ, 2003. *Umělá inteligence: 4*. Praha: Academia. ISBN 80-200-1044-0.
63. MAŘÍK, Vladimír, Olga ŠTĚPÁNKOVÁ a Jiří LAŽANSKÝ, 2013. *Umělá inteligence: 6*. Praha: Academia. ISBN 978-80-200-2276-9.
64. MASTERMAN, Margaret, 1961. Semantic message detection for machine translation, using an interlingua. In: *International Conf. on Machine Translation*. s. 438-475.
65. MCDERMOTT, Drew V., 1976. Artificial intelligence meets natural stupidity. *SIGART Newsletter*.
66. MCGRATH, Andrew. The forum. *ACM SIGGROUP Bulletin* [online]. New York: ACM, 1998, 19(3), 21-25 [cit. 2016-07-31]. DOI: 10.1145/307736.307759. ISSN 23727403. Dostupné z: <http://dl.acm.org/citation.cfm?doid=307736.307759>
67. MINEAU, Guy W, Bernard MOULIN a John F. SOWA, 1993. *Conceptual Graphs for knowledge representation: Proceedings*. Berlin: Springer-Verlag. Lecture notes in artificial intelligence.
68. MINEAU, Guy W, 1994. Views, mappings and functions: Essential definitions to the conceptual graph theory. In: *Conceptual Structures: Current Practices: Second International Conference on Conceptual Structures, ICCS'94 College Park, Maryland, USA August 16-20, 1994 Proceedings*. Berlín: Springer, 160-174. DOI: 10.1007/3-540-58328-9\_11. ISBN 978-3-540-38675-9. Dostupné také z: [http://link.springer.com/10.1007/3-540-58328-9\\_11](http://link.springer.com/10.1007/3-540-58328-9_11)
69. MINEAU, Guy, Gerd STUMME a Rudolf WILLE, 1999. Conceptual Structures Represented by Conceptual Graphs and Formal Concept Analysis. In: TEPFENHART, William M. a Walling CYRE. *Conceptual Structures: Standards and Practices: 7th International Conference on Conceptual Structures, ICCS'99 Blacksburg, VA, USA, July 12-15, 1999 Proceedings*. Berlin: Springer, 423 - 441. DOI: 10.1007/3-540-48659-3\_27. ISBN 978-3-540-48659-6. Dostupné také z: [http://link.springer.com/10.1007/3-540-48659-3\\_27](http://link.springer.com/10.1007/3-540-48659-3_27)
70. MOREAU, Nicolas, Michel LECLÈRE, Michel CHEIN a Alain GUTIERREZ, 2007. Formal and graphical annotations for digital objects. In: *Proceedings of the 2007 international workshop on Semantically aware document processing and indexing - SADPI '07*. New York, New York, USA: ACM Press, s. 69-78. DOI: 10.1145/1283880.1283893. ISBN 9781595936684. Dostupné také z: <http://dl.acm.org/citation.cfm?doid=1283880.1283893>
71. MORTON, S, 1987. *Conceptual Graphs and Fuzziness in Artificial Intelligence*. Bristol. Disertace. University of Bristol.

72. MOTALOVÁ, Tereza a Kateřina PROKOPOVÁ, 2014. Závislostní gramatika. In: *Encyklopedie lingvistiky* [online]. Olomouc: Univerzita Palackého v Olomouci. [cit. 2016-07-30]. Dostupné z: [http://oltk.upol.cz/encyklopedie/index.php5/Z%C3%A1vislostn%C3%AD\\_gramatika](http://oltk.upol.cz/encyklopedie/index.php5/Z%C3%A1vislostn%C3%AD_gramatika)
73. NGUYEN, P.H.P. a D. CORBETT, 2006. A basic mathematical framework for conceptual graphs. *IEEE Transactions on Knowledge and Data Engineering*. 18(2), 261-271. DOI: 10.1109/TKDE.2006.18. ISSN 1041-4347. Dostupné také z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1563987>
74. OBITKO, Marek, 2007. *Ontologies and Semantic Web* [online]. [cit. 2016-07-30]. Dostupné z: <http://www.obitko.com/tutorials/ontologies-semantic-web/ontologies.html>
75. ORDOÑEZ-SALINAS, Sonia a Alexander GELBUKH, 2010. Information Retrieval with a Simplified Conceptual Graph-Like Representation. In: *Advances in Artificial Intelligence: 9th Mexican International Conference on Artificial Intelligence, MICAI 2010, Pachuca, Mexico, November 8-13, 2010, Proceedings, Part I*. Berlin: Springer, s. 92-104. DOI: 10.1007/978-3-642-16761-4\_9. ISBN 978-3-642-16761-4. ISSN 0302-9743. Dostupné také z: [http://link.springer.com/10.1007/978-3-642-16761-4\\_9](http://link.springer.com/10.1007/978-3-642-16761-4_9)
76. PETERSEN, Ulrik, 2005. *Online Course in Knowledge Representation using Conceptual Graphs* [online e-kurz]. Aalborg University, Humanistic Informatics at Department of Communication. [cit. 2015-01-20]. Dostupné z: <http://cg.huminf.aau.dk/>
77. POLOVINA, Simon, 1997. *Assessing Sowa's Conceptual Graphs for Effective Strategic Management Decisions, Based on a Comparative Study with Eden's Cognitive Mapping* [online]. [cit. 2016-07-31]. Dostupné z: <http://homepages.gold.ac.uk/polovina/publications/iccs97paper/>
78. POLOVINA, Simon, 2007. An Introduction to Conceptual Graphs. In: *Conceptual Structures: Knowledge Architectures for Smart Applications: Proceeding ICCS '07 Proceedings of the 15th international conference on Conceptual*. Berlin, Heidelberg: Springer, s. 1. DOI: 10.1007/978-3-540-73681-3\_1. ISBN 978-3-540-73680-6. Dostupné také z: [http://link.springer.com/10.1007/978-3-540-73681-3\\_1](http://link.springer.com/10.1007/978-3-540-73681-3_1)
79. POTTER, Stephen, 2001. *A Survey of Knowledge Acquisition from Natural Language 1* [online]. Edinburgh: University of Edinburgh, (2001) [cit. 2016-07-31]. Dostupné z: <http://www.aii.ed.ac.uk/project/akt/work/stephenp/TMA%20of%20KAfromNL.pdf>
80. QUILLIAN, M. Ross, 1968. *Semantic Memory*. Cambridge: Carnegie Institute of Technology. Dostupné také z: <http://www.dtic.mil/dtic/tr/fulltext/u2/641671.pdf>
81. RAIMBAULT, Thomas, 2012. Transitions-Based System. In: *Foundations of intelligent systems: proceedings of the Sixth International Conference on Intelligent Systems and Knowledge Engineering, Shanghai, China, Dec 2011 (ISKE2011)* [online]. 1. Berlin: Springer, s. 396-405 [cit. 2016-07-31]. DOI: 10.3233/978-1-61499-105-2-396. ISBN 9783642256646. ISSN 1867-5662. Dostupné z: <http://www.springer.com/us/book/9783642256639>
82. Rich, E, 1979. *User Modeling via stereotypes*. Cognitive Science. Vol.3, s. 329-354. ISSN 0364-0213.
83. RICHENS, Richard H, 1956. Preprogramming for mechanical translation. *Mechanical Translation*. 3(1), 20 - 25. Dostupné také z: <http://mt-archive.info/MT-1956-Richens.pdf>
84. RUDOLPH, Sebastian, Markus KRÖTZSCH a Pascal HITZLER, 2007. Quo Vadis, CS? – On the (non)-Impact of Conceptual Structures on the Semantic Web. In: *Conceptual Structures: Knowledge Architectures for Smart Applications: Proceeding ICCS '07 Proceedings of the 15th international conference on Conceptual Structures: Knowledge Architectures for Smart Applications* [online]. Berlin, Heidelberg: Springer Berlin Heidelberg, s. 464-467 [cit. 2016-07-31]. DOI: 10.1007/978-3-540-73681-3\_35. ISBN 978-3-540-73680-6. Dostupné z: [http://link.springer.com/10.1007/978-3-540-73681-3\\_35](http://link.springer.com/10.1007/978-3-540-73681-3_35)
85. SCHANK, Roger C, 1969. A conceptual dependency parser for natural language. In: *COLING '69 Proceedings of the 1969 conference on Computational linguistics*. Stroudsburg: Association for Computational Linguistics Stroudsburg, s. 1-3. DOI: 0.3115/990403.990405.

86. SIDDIQUI, Tanveer J., 2006. Intelligent techniques for effective information retrieval: (a conceptual graph based approach). *ACM SIGIR Forum*. New York: ACM, 40(2).
87. SINGH, Neha, Clark HU a Wesley S. ROEHL, 2007. Text mining a decade of progress in hospitality human resource management research: Identifying emerging thematic development. *International Journal of Hospitality Management* [online]. Elsevier, 26(1), 131-147 [cit. 2016-07-31]. DOI: 10.1016/j.ijhm.2005.10.002. ISSN 02784319. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S027843190500099X>
88. SMITH, Bradley, 1991. CAMES - Expert System Administration of Money Market Services. In: *Proc Conceptual Graphs Workshop, European Conference on AI* (. New York: State University of New York, s. 401-416.
89. SOWA, John F, 1976. Conceptual graphs for a database interface. *IBM Journal of Research and Development*. 20(4), 336-357.
90. SOWA, John F, 1984. *Conceptual Structures: Information Processing in Mind and Machine*. Reading: Addison-Wesley Publishing Company, 1984.
91. SOWA, John F, 1992. Conceptual graphs as a universal knowledge representation. *Computers & Mathematics with Applications* [online]. Elsevier, 23(2-5), 75-93 [cit. 2016-07-30]. DOI: 10.1016/0898-1221(92)90137-7. ISSN 08981221. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/0898122192901377>
92. SOWA, John F, 1993. Logical foundations for representing object-oriented systems. *Journal of Experimental*. 5(2-3), 237-261. DOI: 10.1080/09528139308953772. ISSN 0952-813x. Dostupné také z: <http://www.tandfonline.com/doi/abs/10.1080/09528139308953772>
93. SOWA, John F, 1999. Conceptual Graphs: Draft Proposed American National Standard. In: *Conceptual Structures: Standards and Practices: 7th International Conference on Conceptual Structures, ICCS'99 Blacksburg, VA, USA, July 12-15, 1999 Proceedings*. Berlin: Springer, s. 1-65. DOI: 10.1007/3-540-48659-3\_1. ISBN 978-3-540-48659-6. ISSN 0302-9743. Dostupné také z: [http://link.springer.com/10.1007/3-540-48659-3\\_1](http://link.springer.com/10.1007/3-540-48659-3_1)
94. SOWA, John F, 2000. *Knowledge Representation: Logical, Philosophical, and Computational Foundations*. Pacific Grove: Brooks / Cole. ISBN 0-534-94965-7.
95. SOWA, John F, 2001. Conceptual Graph Standard: ISO International Standard committee draft. In: *Committee on Information Interchange and Interpretation* [online]. 2. 2001-04-02 [cit. 1505]. Dostupné z: <http://users.bestweb.net/~sowa/cg/cgstand.htm>
96. SOWA, John F, 2008. Conceptual Graphs. *Handbook of Knowledge Representation*. F. van Harmelen, V. Lifschitz, B. Porter. Elsevier, 213-237. ISBN 978-0-444-52211-5.
97. SOWA, John F, 2015. *Semantic Networks* [online]. [cit. 2016-07-30]. Dostupné z: <http://www.jfsowa.com/pubs/semnet.htm>
98. STRAWSON, P.F, 1990. *Individuals: an essay in descriptive metaphysics* [online]. London: Routledge, 255 s. [cit. 2016-07-30]. ISBN 0-203-22130-3.
99. STUDERT, Rudi (ed.), 2007. *Semantic web services: concepts, technologies, and applications* [online]. Berlin: Springer. [cit. 2016-07-31]. ISBN 978-3-540-70893-3.
100. ŠTĚPÁN, Jan, 2011. *Logika a právo*. 3. vyd. V Praze: C.H. Beck. Beckovy příručky pro právní praxi. ISBN 978-80-7400-373-8.
101. ŠVEJDA, Jan, 2003. Vyhledávání informací. In: *KTD: Česká terminologická databáze knihovnictví a informační vědy (TDKIV)* [online]. Praha : Národní knihovna ČR, 2003- [cit. 2016-07-31]. Dostupné z: [http://aleph.nkp.cz/F/?func=direct&doc\\_number=000001825&local\\_base=KTD](http://aleph.nkp.cz/F/?func=direct&doc_number=000001825&local_base=KTD).



102. TEPFENHART, W. a C. WALLING, 1999. *Conceptual structures: Standards and practices*. Berlin: Proceedings Berlin: Springer. 7th international conference on Conceptual structures, ICCS'99, Blacksburg, VA, USA, July 12-15.
103. TESNIÈRE, Lucien, 1959. *Éléments de Syntaxe structurale*. 2.vyd. Paris: Librairie C. Klincksieck.
104. TIROSHI, Amit, 2012. Graph based user modeling: Theory and Practice. In: *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces - IUI '12*. New York, New York, USA: ACM Press, s. 11. DOI: 10.1145/2166966.2167051. ISBN 978-1-4503-1048-2. Dostupné také z: <http://dl.acm.org/citation.cfm?doid=2166966.2167051>
105. TOGNINI-BONELLI, Elena, 2001. *Corpus Linguistics at Work*. Amsterdam: John Benjamins, 224 s. ISBN 9789027222763.
106. TRUONG, Quoc-Dinh, Dkak DKAK, Josiane MOTHE a Pierre-Jean CHARREL, 2008. Information Retrieval Model based on Graph Comparison. In: *JADT 2008 : 9es Journées internationales d'Analyse statistique des Données Textuelles*. Dostupné také z: [https://www.irit.fr/publis/SIG/2008\\_JADT\\_TDMC.pdf](https://www.irit.fr/publis/SIG/2008_JADT_TDMC.pdf)
107. VLČEK, Miroslav a Miroslav VACURA, 2014. *Úvod do filosofie společnosti a dějin: antologie z děl vybraných myslitelů*. Vyd. 2., přeprac. Praha: Oeconomica, 2014. ISBN 978-80-245-2039-1.
108. VOLNÁ, Eva a Martin KOTYRBA, 2013. *Umělá inteligence* [online]. Ostrava: Ostravská univerzita. [cit. 2016-07-31]. ISBN 978-80-7464-330-9. Dostupné z: [http://projekty.osu.cz/svp/opory/PrF\\_Volna%2CKotyriba\\_Umela-intelig.pdf](http://projekty.osu.cz/svp/opory/PrF_Volna%2CKotyriba_Umela-intelig.pdf)
109. WAY, Eileen C., 1994. Conceptual graphs — Past, present, and future. In: *Conceptual Structures: Current Practices: Second International Conference on Conceptual Structures, ICCS'94 College Park, Maryland, USA August 16–20, 1994 Proceedings*. Berlín: Springer Berlin Heidelberg, 11 - 39. DOI: 10.1007/3-540-58328-9\_2. ISSN 978-3-540-38675-9. Dostupné také z: [http://link.springer.com/10.1007/3-540-58328-9\\_2](http://link.springer.com/10.1007/3-540-58328-9_2)
110. WEISSTEIN, Eric W, 2016. Isomorphism. In: *MathWorld: A Wolfram Web Resource* [online]. Wolfram. [cit. 2016-07-30]. Dostupné z: <http://mathworld.wolfram.com/Isomorphism.html>
111. WILLE, Rudolf, 2005. Contextual Logic and Aristotle's Syllogistic. In: *Conceptual Structures: Common Semantics for Sharing Knowledge: 3th International Conference on Conceptual Structures, ICCS 2005, Kassel, Germany, July 17-22, 2005. Proceedings*. Berlin: Springer, 238 -249. DOI: 10.1007/11524564\_16. Dostupné také z: [http://link.springer.com/10.1007/11524564\\_16](http://link.springer.com/10.1007/11524564_16)
112. WOODS, William A, 1975. What's in a link: foundations for semantic networks. *Representation and Understanding*. G. Bobrow, A. Collins. New York: Academic Press, s. 35-82.
113. WOOLDRIDGE, Michael, 1995. Intelligent Agents: Theory and Practice. *Knowledge Engineering Review* [online]. 10(2), 115-152 [cit. 2016-07-31]. Dostupné z: <http://www.cs.ox.ac.uk/people/michael.wooldridge/pubs/ker95.pdf>
114. WUWONGSE, Vilas a Mario MANZANO, 1993. Fuzzy conceptual graphs. In: *Conceptual Graphs for Knowledge Representation: First International Conference on Conceptual Structures, ICCS'93 Quebec City, Canada, August 4–7, 1993 Proceedings*. 1. Berlin: Springer, s. 430-449. DOI: 10.1007/3-540-56979-0\_24. ISBN 978-3-540-47848-5. Dostupné také z: [http://link.springer.com/10.1007/3-540-56979-0\\_24](http://link.springer.com/10.1007/3-540-56979-0_24)
115. YANG, Shih-Yao a Von-Wun SOO, 2012. Extract conceptual graphs from plain texts in patent claims. *Engineering Applications of Artificial Intelligence* [online]. Tarrytown: Pergamon Press, 25(4), 874-887 [cit. 2016-07-31]. DOI: 10.1016/j.engappai.2011.11.006. ISSN 09521976. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S095219761100217X>
116. YAO, Haining a Letha ETZKORN, 2006. Automated conversion between different knowledge representation formats. *Knowledge-Based Systems* [online]. 19(6), 404-412 [cit. 2016-07-31]. DOI: 10.1016/j.knosys.2006.04.005. ISSN 09507051. Dostupné z: <http://linkinghub.elsevier.com/retrieve/pii/S0950705106000517>

- 117.ZADEH, Lotfi A, 1973. Outline of a New Approach to the Analysis of Complex Systems and Decision Processes. *IEEE Transactions on Systems, Man, and Cybernetics* [online]. IEEE, SMC-3(1), 28-44 [cit. 2016-07-30]. DOI: 10.1109/TSMC.1973.5408575. ISSN 0018-9472. Dostupné z: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5408575>
- 118.ZANG, Liang-Jun, Cong CAO, Ya-Nan CAO, Yu-Ming WU a Cun-Gen CAO, 2013. A Survey of Commonsense Knowledge Acquisition. *Journal of Computer Science and Technology*. 28(4), 689-719. DOI: 10.1007/s11390-013-1369-6. ISSN 1000-9000. Dostupné také z: <http://link.springer.com/10.1007/s11390-013-1369-6>



## Seznam obrázků

Obr. 1.1 Relační graf znázorňující větu „Farmář vlastní a krmí oslíka.“ .....	18
Obr. 1.2 Existenciální graf znázorňující větu „Pokud má farmář oslíka, tak ho krmí.“ .....	20
Obr. 1.3 Konceptuální graf znázorňující větu „Pokud má farmář oslíka, tak ho krmí.“ .....	20
Obr. 1.4 DRS pro větu „Pokud má farmář oslíka, tak ho krmí.“ .....	22
Obr. 2.1 Grafická forma grafu zobrazující větu „Jan jede do Brna autobusem.“ .....	36
Obr. 2.2 Svaz reprezentující vícečetnou dědičnost .....	43
Obr. 2.3 Svaz typů reprezentující obecnou ontologii Johna F. Sowy .....	44
Obr. 2.4 Matice zobrazující dvanáct typů nacházejících se v centru ontologie .....	44
Obr. 2.5 Koncept obsahující referent a koncept bez referentu .....	46
Obr. 2.6 Konceptuální graf s deskriptorem a dvěma literárními referenty.....	50
Obr. 2.7 Lineární a grafické znázornění koreferenčního spojení 1.....	53
Obr. 2.8 Grafické znázornění koreferenčního spojení 2 .....	53
Obr. 2.9 Grafická forma konceptuálního grafu se dvěma zahrnutými grafy .....	54
Obr. 2.10 Grafická forma grafu s monadickým vztahem pro větu: "Je možné, že pták zpívá." .....	58
Obr. 2.11 Grafická forma grafu s dyadickým vztahem pro větu „Jan jede do Brna".....	58
Obr. 2.12 Grafická forma grafu s triadickým vztahem pro větu „ .....	59
Obr. 2.13 Příklad konceptuálního grafu s užitím zahrnutého grafu .....	66
Obr. 2.14 Obrázek nalevo je reprezentován konceptuálním grafem s kontextem.....	67
Obr. 2.15 Grafická forma konceptuálního grafu s negací .....	68
Obr. 3.1 Konceptuální graf obsahující funkci .....	74
Obr. 3.2 Konceptuální graf definující funkci (VĚK) .....	74
Obr. 3.3 Konceptuální graf s konjunktivním typem konceptu a vztahu.....	76
Obr. 3.4 Fuzzy konceptuální graf s fuzzy konjunktivními typy pro větu: " .....	79
Obr. 3.5 Zahrnutý fuzzy konceptuální graf s koreferenčním spojením .....	79
Obr. 3.6 Pravidlo tvorby omezení .....	84
Obr. 3.7 Pravidlo tvorby spojení .....	85
Obr. 3.8 Pravidlo tvorby zjednodušení.....	86
Obr. 3.9 Operace odečtení.....	87
Obr. 3.10 Operace odpojení.....	88
Obr. 3.11 Svaz typů konceptu .....	89
Obr. 3.12 Zdvojení vztahu .....	90
Obr. 3.13 Konceptuální grafy G a H s různou mírou specializace a zobecnění .....	92
Obr. 3.14 Různé druhy projekcí grafů G a H z Obr. 3.13 .....	93
Obr. 3.15 Maximální společná projekce grafu G a H z Obr. 3.13 .....	93
Obr. 3.16 Maximální spojení grafů G a H z Obr. 3.13.....	95
Obr. 4.1 Konceptuální graf reprezentující větu „Petr si vybírá pokročilé funkce.“ .....	115

Obr. 4.2 Konceptuální graf prototypu P1 a P'1 .....	117
Obr. 4.3 Konceptuální graf prototypu P2 a P'2(.....	117
Obr. 4.4 Funkce definující fuzzy množinu obsaženou v referentu konceptu [RYCHLOST_AKCE] .....	118
Obr. 4.5 Konceptuální graf s znázorňující model uživatele Petr .....	118
Obr. 4.6 Konceptuální graf Q reprezentující dotaz .....	119
Obr. 4.7 Příklad konceptuálního grafu uživatele .....	123
Obr. 4.8 Příklad abstraktního konceptuálního grafu s uživatele k .....	124
Obr. 4.9 Funkce příslušnosti definující fuzzy množinu .....	124
Obr. 4.10 Konceptuální graf prototypu P1 .....	125
Obr. 4.11 Konceptuální graf prototypu P2.....	126
Obr. 4.12 Konceptuální graf Q reprezentující dotaz: .....	127
Obr. 4.13 Ontologie konceptů systému NaZahrádce .....	129
Obr. 4.14 Ontologie vztahů systému NaZahrádce .....	130
Obr. 4.15 Funkce příslušnosti konceptu HUMIDITA.....	130
Obr. 4.16 Konceptuální graf reprezentující větu G1: "Rostliny se většinou sázejí na jaře." .....	131
Obr. 4.17 Konceptuální graf reprezentující větu G2: "Růže se vždy sázejí do vlhké půdy." .....	131
Obr. 4.18 Maximální společná projekce P grafu G1 a G2.....	131
Obr. 4.19 Konceptuální graf M: Maximální spojení grafů G1 a G2 .....	132
Obr. 4.20 Konceptuální graf Q reprezentující dotaz: "V jakém ročním období se sázejí růže?" .....	132

## Seznam vzorců

Vzor. 1.1 Piercova notace věty „Farmář vlastní oslíka.“ .....	18
Vzor. 1.2 Notace věty „Farmář vlastní oslíka.“ v predikátovém kalkulu .....	18
Vzor. 1.3 Relační graf znázorňující větu „Farmář vlastní oslíka.“ .....	18
Vzor. 1.4 Notace věty „Farmář vlastní a krmí oslíka.“ v predikátovém kalkulu .....	19
Vzor. 1.5 Notace věty „Pokud má farmář oslíka, tak ho krmí.“ v predikátovém kalkulu .....	21
Vzor. 1.6 Notace věty „Pokud má farmář oslíka, tak ho krmí.“ v predikátovém kalkulu 2 .....	21
Vzor. 2.1 Spojení dvou identických typů konceptu do jednoho konceptuálního grafu .....	33
Vzor. 2.2 Lineární forma zápisu věty: „Jan jede do Brna“ .....	37
Vzor. 2.3 „Existuje tvrzení, které říká, že kočka honí myš.“ .....	49
Vzor. 2.4 Lineární znázornění koreferenčního spojení.....	54
Vzor. 2.5 Lineární forma konceptuálního grafu se dvěma zahrnutými grafy .....	55
Vzor. 2.6 Dva rozdílné zápisy definovaného návěští určeného pomocí n-adických lambda výrazů .....	57
Vzor. 2.7 Lineární zápis věty „Jan jede do Brna“ .....	57
Vzor. 2.8 Rozšířený lineární zápis věty „Jan jede do Brna“ .....	57
Vzor. 2.9 Lineární zápis konceptuálního grafu vyjadřujícího: „Osoba Jan je agentem JET a Město Brno je cílem Jetí.“ .....	57
Vzor. 2.10 Lineární forma grafu s monadickým vztahem pro větu: "Je možné, že pták zpívá." .....	58
Vzor. 2.11 Lineární forma grafu s dyadickým vztahem pro větu „Jan jede do Brna" (Sowa, 1984) .....	58
Vzor. 2.12 Lineární forma grafu s triadickým vztahem pro větu „Osoba je mezi skálou a tvrdým místem.“.....	59
Vzor. 2.13 Využití lambda kalkulu při definici konceptuálních vztahů (JETNĚKAM) a (JETDOBRNA).....	62
Vzor. 2.14 Lineární forma grafu využívající vztah definovaný lambda kalkulem (JETDOBRNA) reprezentující větu „Profesor Červíček jede do Brna“.....	62
Vzor. 2.15 Využití lambda kalkulu pro definici typu konceptu.....	63
Vzor. 2.16 Využití lambda kalkulu pro definici typu konceptuálního vztahu. ....	64
Vzor. 2.17 Příklad konceptuálního grafu s užitím zahrnutého grafu reprezentující větu „Modrý lampión bliká.“ .....	65
Vzor. 2.18 Konceptuální graf s kontextem pro koncept typu [TVRZENÍ] pro větu „Všichni sloni jsou šedí.“ .....	67
Vzor. 2.19 Konceptuální graf s kontextem pro koncept typu [TVRZENÍ] pro větu „Jumbo je šedý slon.“ (Petersen, 2005).....	67
Vzor. 2.20 Lineární forma konceptuálního grafu s negací.....	68
Vzor. 2.21 Lineární forma konceptuálního grafu s konjunkcí .....	68
Vzor. 2.22 Příklad řešení disjunkce DeMorganovým pravidlem .....	69
Vzor. 2.23 Lineární forma konceptuálního grafu s implikací.....	69

Vzor. 3.1 Lineární forma zápisu věty v CGIF: „Jan jede do Brna.“ .....	72
Vzor. 3.2 Lineární forma formule 1.1 zkrácená pomocí zahnížděných grafů .....	72
Vzor. 3.3 Lineární forma zápisu věty v CLIF: „Jan jede do Brna.“ .....	72
Vzor. 3.4 Pro srovnání lineární forma zápisu věty v KIF: „Jan jede do Brna.“ .....	72
Vzor. 3.5 Koncept v časovém konceptuálním grafu .....	80
Vzor. 3.6 Konceptuální graf věty „Neználek by nemohl letět na Mars“ .....	81
Vzor. 3.7 Časové konceptuální grafy .....	81
Vzor. 3.8 Konceptuální graf, který je sémanticky správně, ale neeense žádný význam .....	82