

Univerzita Karlova v Praze

Filozofická fakulta

Katedra logiky



Bakalářská práce

Bc. Petr Šudoma

Významné směry v umělé inteligenci

Important Themas in Artificial Intelligence

Děkuji vedoucímu práce Doc. PhDr. Petru Jirků, CSc. a konzultantovi Mgr. Jonathanu Vernerovi PhD. za cenné rady, připomínky a čas, který mi věnovali.

Petr Šudoma

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně, že jsem řádně citoval 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 19.8.2013

.....

Bc. Petr Šudoma

Abstrakt:

Práce studuje deskripční logiky jako metodu reprezentace znalostí umělé inteligence, popisuje historický vývoj reprezentace znalostí vedoucí k vzniku deskripčních logik a srovnává jejich výhody a nevýhody se systémy rámců, z kterých deskripční logiky vznikly. Práce představuje formální syntaxi a sémantiku, zavádí pojmovou systematiku deskripčních logik a popisuje algoritmy pro nejčastěji používané odvozovací úlohy na deskripčních logikách, pro subsumpci a konzistenci. Práce se zabývá i problematikou výpočtové složitosti těchto algoritmů v závislosti na konkrétní použité deskripční logice. Práce ukazuje výhody i další potenciál deskripčních logik.

Klíčová slova:

umělá inteligence, reprezentace znalostí, inference, deskripční logiky, subsumpce

Abstrakt:

The paper studies description logics as a method of field of artificial intelligence, describes history of knowledge representation as series of events leading to founding of description logics.

Furthermore the paper compares description logics with their predecessor, the frame systems.

Syntax, semantics and description logics naming convention is also presented and algorithms solving common knowledge representation tasks with usage of description logics are described.

Paper compares computational complexity of subsumption of several description logics.

Usefulness and further possibilities of description logic systems are shown.

Keywords:

artificial intelligence, knowledge representation, inference, description logics, subsumption

Obsah

Obsah.....	6
Úvod.....	8
1. Umělá inteligence a reprezentace znalostí.....	9
1.1 Umělá inteligence.....	9
1.2 Reprezentace znalostí.....	10
1.3 Formální jazyky pro reprezentaci znalostí.....	11
2. Historie reprezentace znalostí a deskripčních logik v umělé inteligenci.....	13
2.1 Prehistorie Umělé inteligence a reprezentace znalostí	13
2.2 Počátek Umělé inteligence a reprezentace znalostí	13
2.3. Rozpoznání významu reprezentace znalostí.....	14
2.4 Vznik deskripčních logik.....	15
3. Deskripční logiky.....	16
3.1 Logiky konceptů.....	16
3.2 TBox, RBox a ABox.....	16
3.3 Syntaxe deskripčních logik.....	17
3.3.1 Třídy objektů.....	17
3.3.2 Konstruktory konceptů.....	17
3.3.3 Konstruktory rolí.....	19
3.3.4 Atomické formule.....	19
3.4 Sémantika deskripčních logik.....	19
3.4.1 Interpretace.....	19
3.4.2 Interpretace složených výrazů.....	20
3.4.3 Teorie a modely teorií.....	22
3.5 Konkrétní deskripční logiky.....	22
3.5.1 Jazyk deskripční logiky AL.....	23
3.5.2 Jazyk deskripční logiky FL.....	23

3.5.3 Jazyk deskripční logiky EL.....	24
3.5.4 Rozšíření syntaxe základních jazyků deskripčních logik.....	24
3.5.5 Rozšíření úpravou sémantiky deskripčních logik.....	25
3.5.6 Deskripční logika S.....	25
4. Odvozovací úlohy reprezentace znalostí a deskripční logiky.....	26
4.1 Odvozovací úlohy.....	26
4.2 Vzájemná převoditelnost odvozovacích úloh.....	27
4.3 Algoritmus strukturální subsumpce.....	27
4.4 Algoritmus sémantických tableaux.....	28
4.5 Složitost subsumpce ve vybraných deskripčních logikách.....	30
Závěr.....	32
Seznam použité literatury.....	33

Úvod

Deskripční logiky jsou poměrně moderním a relativně úspěšným formalismem pro reprezentaci znalostí. Vyznačují se jak dostatečnou expresivitou, tak v praxi použitelnou výpočtovou složitostí i pevnými formálními základy. Všechny tyto vlastnosti jsou u dobrého systému pro reprezentaci znalostí nezbytností a deskripční logiky jsou dobře definované logiky s formální sémantikou, umožňující studium jejich teoretických vlastností (expresivita, převoditelnost, výpočtová složitost) i s možností vybrat si z mnoha deskripčních logik takovou, jejíž poměr expresivita/výpočtová složitost odvození bude nejpříznivější pro konkrétní praktickou aplikaci. Jejich kvality je možno hodnotit i zprostředkovaně, podle množství prací, které se jimi zabývá, podle úspěšnosti a počtu znalostních systémů, které jsou na nich postaveny. Rovněž standardizace deskripčních logik jako teoretického základu webových sémantik ukazuje, že deskripční logiky nejsou slepou vývojovou větví, ale že patří k tomu lepšímu mezi tím, co dnes umělá inteligence, reprezentace znalostí i formální logika může používat a nabídnout dalším vědeckým odvětvím k využití.

Tento text představuje deskripční logiky v kontextu reprezentace znalostí a umělé inteligence, zabývá se vztahem reprezentace znalostí a umělé inteligence, mapuje vývoj a důvody vedoucí ke vzniku deskripčních logik v rámci tohoto oboru, srovnává jejich výhody a nevýhody s příbuznou rodinou metod na reprezentaci znalostí – systémy rámců. Formálně zavádí rodinu deskripčních logik v maximální možné šíři a definuje jejich formální syntaxi a sémantiku. Popisuje algoritmy pro nejčastěji používané odvozovací úlohy subsumpce a splnitelnosti na deskripčních logikách a na závěr prezentuje výsledky z oblasti vztahu výpočtové složitosti odvozovacích úloh a expresivity deskripčních logik.

1. Umělá inteligence a reprezentace znalostí

1.1 Umělá inteligence

Umělá inteligence je věda poměrně nová a stále ještě ne dostatečně rozvinutá a problémem každé nové vědy je, že nemá jasně vyvinutou filosofii sebe sama, že nemá přesně definováno, jaký je její objekt zkoumání, jaké jsou její metody práce ani jaká jsou její kritéria hodnocení úspěšnosti. Pokud se zastavíme již u první položky, to jest objektu zkoumání umělé inteligence, měla by jím být inteligence a to inteligence umělá, čímž je myšleno inteligence vytvořená člověkem. Tato triviální a jasná definice nic neřeší, protože navzdory úsilí jiných věd není jasné, ani co je to inteligence ne-umělá, tedy inteligence „přirozená“, a ani co je to inteligence obecně bez přízvisek. Jelikož pojem inteligence nelze použít pro definici objektu zkoumání umělé inteligence, je tedy pro definici objektu zkoumání umělé inteligence nutné zvolit jiný přístup.

Je možné, že vhodným způsobem, jak definovat objekt zkoumání umělé inteligence je následující způsob pragmatický. Stejně jako jiné vědy, se umělá inteligence snaží šetřit lidskou práci. Vědy se snaží šetřit lidskou práci (vykonávat ji místo člověka) tím, že se snaží přesunout nutné činnosti z člověka na nějakého alternativního vykonavatele, pravidelně na nějaký systém fyzický, mechanický či formální. (Alternativně - se vědy snaží získávat znalosti, které takový přesun umožní, či znalosti, které takový přesun usnadní, zefektivní, zefektivní samotou prací apod.) Tento obecný koncept můžeme pozorovat u významné podmnožiny vědních oborů. Lze říci, že umělá inteligence spadá do této kategorie, protože umělá inteligence se snaží šetřit duševní práci přesunem těchto prací na počítač. Umělá inteligence tedy zkoumá metody, které umožní stroji vykonávat různé duševní činnosti za člověka a snaží se o realizaci těchto metod. Ovšem toto lze říci i o klasické informatice, tato definice tedy není postačující. Klasická informatika používá jako svůj základní pracovní prostředek algoritmus, který musí být hlavně konečný a korektní. Umělá inteligence taková omezení obecně nepožaduje a tím se od klasické informatiky liší.

Umělá inteligence je tedy věda, která se snaží o zkoumání a realizaci systémů, které jsou schopné šetřit duševní práci (vykonávat ji místo člověka), systémů, které nejsou realizovatelné pomocí metod klasické informatiky, a o zkoumání a řešení takových duševních prací (úloh), které nejsou řešitelné metodami klasické informatiky.

Do doby, než bude jasné, co to je inteligence, aby bylo možno umělou inteligenci definovat pomocí pojmu inteligence, by takováto definice umělé inteligence mohla být postačující.

1.2 Reprezentace znalostí

Při řešení úloh umělé inteligence se předpokládá, že všechna řešení hlavních úloh umělé inteligence jsou převoditelná na problematiku odvozování nových skutečností ze skutečností již známých. Záhy se ukázalo, že k efektivnímu odvozování je třeba nějakým způsobem zpracovávat znalosti, a že problematika znalostí je pro umělou inteligenci klíčová a centrální, tedy, že většinu klíčových problémů, které mají inteligentní systémy řešit, je možné řešit, pokud řešitel má a umí efektivně využívat znalosti řešení problémů obecně a ideálně i znalosti řešení dané konkrétní třídy problémů.

Je mnoho způsobů jak zařídit, aby řešitel znalosti měl a mohl s nimi pracovat, i tak je ale možné a obvyklé je rozdělit do dvou základních skupin, a sice na implicitní, což znamená, že znalosti jsou zapsány přímo v programu řešitele, a explicitní, to jest reprezentované v nějakém jazyce, který řešitel umí číst a takto zapsané znalosti má k dispozici pro čtení podle své potřeby. Toto obvyklé dělení znalostí na implicitní a explicitní není neproblematické, nicméně postačuje k ilustraci základních výhod a nevýhod těchto přístupů.

Výhodou implicitních znalostí je, že jejich zpracování je vždy rychlejší než zpracování znalostí explicitních, protože odpadá nutnost čtení a interpretace zápisu znalostí nutná u znalostí explicitních. Rovněž může být v některých případech výhodou, když znalosti v implicitní podobě nejsou čitelné člověkem. Jsou situace, kdy člověk vyhodnotí konkrétní znalost stroje jako chybnou a přitom tato znalost podstatným způsobem vylepšuje schopnost stroje plnit zadanou úlohu. Odstranění takové „chybné“ znalosti má pak za následek snížení schopností stroje.

Na druhé straně explicitní znalosti - to jest znalosti zapsané pomocí nějaké explicitní reprezentace - lze snadno číst, kontrolovat, měnit, přidávat či odebrat, neboli jsou flexibilnější. Práce s takovou reprezentací je ale pomalejší, navíc přibývá problém volby reprezentace. Navzdory těmto problémům se obecně explicitní reprezentace znalostí považuje za lepší hlavně kvůli své flexibilitě. Hlavní směr umělé inteligence považuje explicitně reprezentované znalosti jako součást svého paradigmatu.

Explicitní reprezentace znalostí přináší problém volby této reprezentace. Obvykle se pro tento účel používají různé formální jazyky. A právě možné volby formálního jazyka

pro reprezentaci znalostí a vlastnosti jednotlivých formálních jazyků jsou doménou oboru reprezentace znalostí.

1.3 Formální jazyky pro reprezentaci znalostí

Formální jazyky pro reprezentaci znalostí mají primárně umožňovat snadnou inferenci, snadnou čitelnost, mají mít granularitu, umožňující snadné přidávání i odebírání jednotlivých znalostí. Jelikož důvodem pro uložení těchto znalostí je inference, je pochopitelné, že většina forem pro ukládání jsou různé formální logiky. Nicméně některé poměrně úspěšné systémy pro reprezentaci znalostí nebyly postaveny na logickém základě a přestože logikami ve své podstatě byly, v literatuře a úvahách o nich se neužíval primárně logický diskurs ani metody.

Problémem takových systémů jsou právě chybějící logické základy, což znemožňuje využití již vyzkoušených inferenčních algoritmů používaných pro různé formální logiky. A nejen to, tím, že danému systému reprezentace znalostí chybí logické základy, je znemožněno nebo minimálně značně znesnadněno používání již velice širokého spektra lidských znalostí, týkajících se inference v různých logických systémech, ať se již jedná o konkrétní algoritmy, odhady složitostí jednotlivých algoritmů pro různé třídy logik nebo i praktické zkušenosti z používání takových algoritmů. Jinými slovy, je nutno pro takový systém inferenční algoritmy vyvíjet zcela znovu nebo přinejmenším více či méně komplikovaně upravovat již známé algoritmy pro daný konkrétní systém. Dalším problémem, který absence logických a obecně formálních základů u systémů, snažících se o inferenci, přináší, je faktické znemožnění vylepšování a rozšiřování systému. Pokud není systém dostatečně formálně popsán a inferenční algoritmus je bez teoretických základů, je prakticky nemožné takový systém rozšířit. Není totiž zřejmé, zda se někde v systému v některém z jeho ad-hoc odvozovacích algoritmů nevyskytují implicitně použité předpoklady, které budou rozšířením falsifikovány, a v důsledku této falsifikace nezačne rozšířený systém dávat chybné výsledky. Zpětná kontrola takových situací nebo jejich prevence je u ad-hoc systémů řádově obtížnější, ne-li nemožná. Výsledkem potom je, že když není možné systémy snadno rozšiřovat, je nutné navrhovat nové systémy víceméně od začátku, a protože není jasné, jaké části již hotové práce mohou být použity, je nutné některé části práce duplikovat.

Mezi takové metody a systémy nepostavené na logickém základě, které trpěly předchozími nedostatky, patří rámce (Minsky 1974) a z nich odvozené systémy. Na druhou stranu, tyto systémy byly v praxi použitelné, protože v nich bylo možné vybalancovat složitost odvo-

zení a expresivitu jazyka takovým způsobem, že byly dostatečně rychlé.

Tyto výhody se povedlo opakovaně prokázat konstrukcí různých systémů založených na rámcích (např. Lederberg 1987).

Z tohoto hlediska je pochopitelné, že po úspěších, kterých dosáhly systémy založené na rámcích, se začalo hledat řešení, které by spojilo výhody rámců a formálních logik. Takový systém by měl mít formálně definovanou syntaxi a sémantiku na úrovni, na jakou jsme zvyklí z formálních logik, ale i expresivní sílu rámců a zároveň prakticky použitelnou složitost odvozování.

Zatím se zdá, že takovým systémem by mohly být právě deskripční logiky, jsou expresivitou velice podobné rámcům, jako dobře definované formální logiky mají pevné formální základy a dobře definovanou syntax i sémantiku. Z toho pak plyne, že se dají dobře kategorizovat mezi ostatní známé logiky, a tedy je jasné, jaké již známé metody a postupy je možno na ně použít rovnou bez změn a jaké je třeba upravovat. Navíc jsou v poslední době velmi intenzivně zkoumány a množství znalostí ohledně složitosti jejich inference roste velmi rychle.

2. Historie reprezentace znalostí a deskripčních logik v umělé inteligenci

Jako každá věda, i umělá inteligence nemá jednoznačně určitelný jeden počátek své historie. V historii, která k umělé inteligenci vedla, existuje mnoho myšlenek, konceptů, článků, knih i událostí, které jsou pro umělou inteligenci natolik významné, že je možné je za počátek z nějakého důvodu označit. Nicméně nejčastěji je za počátek umělé inteligence jako vědy považována konference z roku 1956, konaná na Dartmouth College v Hannoveru v New Hampshire. Ale i v období před touto konferencí je možno najít myšlenky, koncepty, které ke vzniku umělé inteligence a reprezentace znalostí jako nového vědního odvětví vedly a které by měly být považovány za jeho součást.

2.1 Prehistorie Umělé inteligence a reprezentace znalostí

První komplikovaná intelektuální činnost, kterou zvládl vykonávat stroj, je sčítání a obvykle se uvádí, že prvním, kdo zvládl takový stroj sestavit, byl roku 1642 Blaise Pascal. Pascalova sčítačka (obvykle známá pod jménem Pascalina) byla osmimístná, z toho dvě místa byla brána jako desetinná, tedy šlo již o stroj, který byl schopen sčítat netriviálně velká čísla a šetřit skutečnou práci. Pascalina jako taková není součástí historie umělé inteligence ani reprezentace znalostí, i když je bezpochyby jedním z velkých milníků informatiky. Zásadní význam Pascaliny pro umělou inteligenci spočívá v tom, že se jejími vlastnostmi zabýval Gottfried Wilhelm Leibniz. Leibniz se mechanickými počítacími stroji zabýval, dokonce podstatným způsobem přispěl k jejich budoucímu rozvoji vynálezem Leibnizova kola. Avšak hlavní přínos pro umělou inteligenci je možno spatřovat v jeho konceptu a myšlence univerzálního formálního jazyka a inferenčního kalkulu nad tímto jazykem. Protože toto své dílo nikdy nedokončil a tato část Leibnizova díla není v popředí zájmu historiků, je stále otevřeným problémem, zda Leibniz zamýšlel pouze sestavit inferenční kalkul nebo i sestavit inferenční kalkulátor. V každém případě po sobě zanechal koncept či ideu obecného jazyka na reprezentaci znalostí a inferenčního algoritmu nad tímto jazykem, což je princip všech současných systémů pracujících se znalostmi. Na Leibnizovu práci navázal až Gottlob Frege, který realizoval Leibnizovu myšlenku jazyka a inferenčního kalkulu, které sice nebyly samy o sobě univerzální, ale podnítily výzkum a snahy o takové jazyky a eventuálně kalkuly.

2.2 Počátek Umělé inteligence a reprezentace znalostí

Dartmoutská konference, celým oficiálním jménem Dartmouth Summer Research Project on Artificial Intelligence, je událost obecně považovaná za začátek historie umělé inteligence

a hlavně její tradiční větve, kam patří i reprezentace znalostí. V rámci této konference, které se účastnili i Allen Newell a Herbert Alexander Simon, byl předveden i Logic Theorist, což byl první program, který nějakým způsobem pracoval se znalostmi a byl schopen dokazovat matematické věty, a to dokonce tak dobře, že našel elegantnější a kratší alternativu k jednomu již známému matematickému důkazu. Pro reprezentaci znalostí používal výrokovou logiku se spojkami negace, implikace a disjunkce. Jeho úkolem bylo dokázat dokazatelnost tvrzení z pevných axiomů za použití odvozovacích pravidel substituce a Modus Ponens (Newell, Simon 1956).

Následovníkem programu Logic Theorist byl program General Problem Solver, považovaný za program ustavující paradigmatu oboru reprezentace znalostí. Měl oddělenou programovou část od odvozovacích pravidel, takže jako takový je možno jej vnímat jako první produkční systém. Vstupem byla množina odvozovacích pravidel, popis problému a množina cílových stavů. Na GPS je možno nahlížet jako na program, který prohledává prostor, z aktuálního stavu se může dostat do n stavů, určených použitelnými aplikacemi produkčních pravidel; z těchto si vybírá podle heuristiky takový krok, který jej o největší krok posune k řešení problému.

Dalším důležitým konceptem nebo hypotézou, která je pro obor reprezentace znalostí klíčová je idea, že lidské intelektuální činnosti se dají redukovat na manipulaci se symboly nějakého jazyka. Newell se Simonem tuto hypotézu formulovali (Newell, Simon 1976) po jejich úspěchu s GPS. Bez přijetí této hypotézy není možné uvažovat žádný současný systém pro reprezentaci znalostí.

2.3. Rozpoznání významu reprezentace znalostí

První návrhy systémů umělé inteligence ignorovaly výpočtovou složitost užívaných algoritmů a pravidelně selhávaly na větších problémech díky kombinatorické explozi. Tento problém se podařilo obejít sestavením systémů, jejichž ústřední částí byla databáze specifických znalostí o problému.

Takovým systémem byl DENDRAL, specializovaný program na identifikaci molekulární struktury na základě dat hmotnostního spektrometru. DENDRAL byl modulární systém, obsahující databázi znalostí ve formě produkčních pravidel, který byl pozoruhodný i tím, že byl schopen sám z výsledků experimentů navrhnout nová produkční pravidla a tím se učit. Je pozoruhodné, že takto získané znalosti (ve formě pravidel) byly zajímavé i pro odborné chemiky

a DENDRAL je jeden z prvních systémů reprezentace znalostí, který, nejenže znalosti přijímal, ale dokonce i vydával nové.

2.4 Vznik deskripčních logik

V letech 1965-1980 vznikly v reprezentaci znalostí dva významné systémy, které předznamenávaly nástup deskripčních logik. Byly to sémantické sítě a systémy rámců. Sémantické sítě jsou systémem na reprezentaci znalostí, založený na grafové struktuře, kde dnešní terminologií řečeno individua jsou vrcholy a ohodnocené hrany vyjadřují vzájemné vztahy těchto individuí. Rámce (Minsky 1974), ze kterých se mimo jiné vyvinulo dnešní objektově-orientované programování, jsou individua svázaná vztahy specializace/generalizace s možností instanciace. Vliv rámců je na deskripčních logikách velmi dobře viditelný, jsou v podstatě přímým předkem deskripčních logik a dá se říci, že deskripční logiky bez formální sémantiky jsou rámce.

Klíčové pro deskripční logiky je období 1980 – 1990, kdy vznikla první generace deskripčních logik a systémů na nich založených. Nejvýznamnějším a nejznámějším je systém KL-ONE (Brachman, Schmoltze 1985). Většinou se jednalo o deskripční logiky málo expresivní, využívající nějaké varianty algoritmů strukturální subsumpce. V tomto období se rovněž objevují první poznatky o složitosti deskripčních logik, konkrétně o zajímavém jevu, kdy s malým nárůstem expresivní síly jazyka velmi výrazně naroste výpočtová složitost odvozovacích úloh. Objev tohoto chování byl pro deskripční logiky velmi významný. Možnost zkoumání vlivu tak drobné změny, jako je přidání jednoho konstruktora, vyústící v zásadní změnu expresivity jazyka, je to, co činí deskripční logiky přitažlivými z teoretického hlediska.

V dalších pěti letech se začaly v deskripčních logikách používat algoritmy, které byly založené na sémantických tableaux, a s nimi přichází zájem o expresivnější deskripční logiky, protože se ukazuje, že navzdory poměrně vysokým teoretickým horním mezím výpočtové složitosti odvozovacích úloh jsou i tyto expresivní deskripční logiky použitelné a odvozování na nich je v praxi dostatečně rychlé, pokud se věnuje dostatečná pozornost optimalizacím programu.

Nyní jsou podrobně zkoumány možnosti převodu úloh z a do deskripčních logik, z a do logik modálních a do fragmentů prvořákové logiky. Zároveň jsou již k dispozici poměrně velké databáze znalostí zapsané v různých deskripčních logikách. Nicméně nejviditelnější a nejvíce popularizující událostí pro deskripční logiky bylo jejich přijetí jako formalismu pro jazyky webových ontologií OWL a jejich standardizace organizací W3C.

3. Deskripční logiky

3.1 Logiky konceptů

Deskripční logiky jsou primárně určeny k zápisům ontologií, tudíž jejich ústředním tématem jsou vzájemné vztahy pojmů (např. vztah mezi pojmem „člověk“ a „neandrtálec“), definování pojmů z jiných pojmů, vztahy pojmů a jednotlivých individuí a v menší míře i popis relací mezi individuí. Ze svých historických kořenů v systémech rámců a sémantických sítí převzaly deskripční logiky i názvosloví, takže u deskripčních logik nehovoříme o pojmech, nýbrž o konceptech, a nehovoříme o vztazích individuí, nýbrž o rolích.

Pojem a idea konceptu je v deskripčních logikách ústřední. Největší část syntaxe deskripčních logik se zabývá právě koncepty, především utvářením složených konceptů z jednodušších nebo i utvářením konceptů z rolí. Větší část syntaxe formulí v deskripčních logikách pak také umožňuje mluvit o konceptech, nejčastěji o možnosti koncepty porovnávat.

3.2 TBox, RBox a ABox

Jelikož rozhodovací úlohy o vztazích mezi koncepty nepotřebují (ve většině deskripčních logik) ke svému běhu tvrzení o příslušnosti individuí jednotlivým konceptům, používá se dělení vstupních, zadaných nebo v znalostní bázi obsažených axiomů nebo tvrzení na dvě nebo někdy i tři části: Na část tvrzení o individuích, na část, která hovoří o vztazích mezi koncepty a někdy i na část která hovoří o vztazích mezi rolemi.

Pro část tvrzení nebo axiomů hovořící o individuích je zaveden název ABox (z *angl.* membership Assertion, protože většina tvrzení o individuích je prohlášení o příslušnosti individuí do konceptů a rolí), zatímco pro druhou část, hovořící pouze o vztazích termínů (konceptů) se používá název TBox (z *angl.* Terminological knowledge). Některé bohatší logiky mohou obsahovat syntaxi pro definování vzájemných vztahů rolí, množinu takových axiomů nazýváme RBox.

Pro některé odvozovací úlohy postačuje jako vstup TBox, tj. nezávisí na ABoxu, což je praktický důvod pro rozdělování axiomů deskripčních logik. Například úloha ověřit, zda definice konceptu je splnitelná (tj. není vnitřně sporná), nebo úloha tzv. „subsumpce“ (tj. ověření, zda jsou dva koncepty podle své definice ve vztahu generalizace/specializace,) nejsou závislé na axiomech ABoxu. Nicméně ABox potřebujeme například pro úlohu ověření, zda celá znalostní báze má model.

3.3 Syntaxe deskripčních logik

Protože deskripční logiky jsou rodinou logik a protože vzájemně sdílejí mnoho své syntaxe, je nevýhodné definovat každou deskripční logiku zvlášť, je výhodnější, uvést většinu syntaktických konstruktů na jednom místě a jednotlivé deskripční logiky pak definovat určením, jaká podmnožina syntaktických konstruktů je přípustná v které logice. V této sekci budou zavedeny syntaktické konstrukty a v dalších sekcích jim bude nejdříve definována formální sémantika, a posléze budou pomocí takto definovaných částí sestaveny konkrétní deskripční logiky.

3.3.1 Třídy objektů

Deskripční logiky pracují se čtyřmi druhy objektů a gramatika deskripčních logik bude definovat čtyři kategorie výrazů: individua (tj. individua univerza), koncepty (tj. unární relace na individuích univerza), role (tj. binární relace na individuích univerza) a sentence (o příslušnosti individuí do konceptů a rolí, o vzájemných vztazích konceptů, o vzájemných vztazích rolí a o vzájemných vztazích konceptů a rolí).

Jazyk deskripčních logik obsahuje symboly pro jména individuí (množinu jmen pro individua značíme N_I), symboly pro jména konceptů (množinu jmen konceptů značíme N_C), symboly pro jména rolí (množinu jmen rolí značíme N_R). Dále obsahuje logické symboly pro konstruktory konceptů a rolí. Množina použitých logických symbolů závisí na konkrétní deskripční logice.

3.3.2 Konstruktory konceptů

Většina syntaktických konstrukcí deskripčních logik umožňuje zapsat definici složitějšího konceptu z jednodušších pomocí některého z operátorů. Tyto operátory, které slouží pro definice složených konceptů, se nazývají konstruktory konceptů. Konstruktory konceptů mohou tvořit definici nového konceptu z konceptu, z dvojice konceptů, z konceptu a role nebo i pomocí prostého výčtu individuí z univerza. Rovněž se v některých deskripčních logikách vyskytují „konceptové konstanty“ univerzální a prázdný koncept. V této kapitole zadefinujeme syntaxi nejčastěji se vyskytujícími konstruktorů konceptů českých a anglických názvů jednotlivých konceptů (pokud jsou tyto ustálené).

Nechť jsou a_1, \dots, a_n jména individuí, A atomický koncept, C a D jsou koncepty a R je role, pak platí následující:

Koncept výčtem prvků (*angl.* concept by enumeration):

Pokud jsou a_1, \dots, a_n jména individuí, potom $\{a_1, \dots, a_n\}$ je koncept.

Univerzální koncept (*angl.* universal concept) a prázdný koncept (*angl.* bottom concept):

\top a \perp jsou koncepty.

Koncept atomickým doplňkem (*angl.* atomic negation) a koncept doplňkem (*angl.* negation):

Pokud je A atomický koncept a C koncept, potom $\neg A$ a $\neg C$ jsou koncepty.

Koncept sjednocením (*angl.* union) a koncept průnikem (*angl.* intersection):

Pokud jsou C, D koncepty, potom $C \sqcup D$ a $C \sqcap D$ jsou koncepty.

Koncept slabou existenční kvantifikací (*angl.* limited existential quantification):

Pokud je R role, potom $\exists R. \top$ je koncept.

Koncept slabým omezením počtu zdola (*angl.* at-least number restriction):

Pokud je R role, potom $\geq n R$ je koncept.

Koncept omezením počtu shora (*angl.* at-most number restriction):

Pokud je R role, potom $\leq n R$ je koncept.

Koncept omezením počtu rovností (*angl.* exact number restriction):

Pokud je R role, potom $=n R$ je koncept.

Koncept silnou existenční kvantifikací (*angl.* existential quantification):

Pokud je C koncept a R role, potom $\exists R.C$ je koncept.

Koncept obecnou kvantifikací (*angl.* value restriction nebo i role restriction)

Pokud je C koncept a R role, potom $\forall R.C$ je koncept.

Koncept slabým omezením počtu zdola (*angl.* qualified at-least number restriction):

Pokud je C koncept a R role, potom $\geq n R.C$ je koncept.

Koncept omezením počtu shora (*angl.* qualified at-most number restriction):

Pokud je C koncept a R role, potom $\leq n R.C$ je koncept.

Koncept omezením počtu rovností (*angl.* qualified exact number restriction):

Pokud je C koncept a R role, potom $=n R.C$ je koncept.

3.3.3 Konstruktory rolí

V některých expresivnějších deskripčních logikách je možné definovat složitější role pomocí jednodušších, rovněž i pro tyto konstruktory zde definujeme syntax. Opět, stejně jako v předchozím odstavci, definujeme nejčastěji se vyskytující konstruktory. Konstruktory rolí nejsou centrální oblastí zájmu, vyskytují se jen v některých expresivnějších deskripčních logikách, a proto nemají zažité anglické pojmenování. Navíc konstruktory rolí většinou triviálně kopírují základní množinové operace, takže speciální pojmenování by byla nadbytečná.

Necht' jsou R a S role a C koncept, pak platí následující: U , R^- , R^+ , R^ , $\neg R$, $R \sqcup S$, $R \sqcap S$, $R \circ S$, $id(C)$ a $R|_C$ jsou role.*

3.3.4 Atomické formule

Z formulí se v deskripčních logikách vyskytují pouze formule atomické. V ostatních logikách se syntaktická pestrost projevuje na úrovni formulí, v podobě množství operátorů nebo kvantifikátorů. V deskripčních logikách takové operátory nejsou zapotřebí, deskripční logiky jsou primárně logiky konceptů, syntaktická pestrost se projevuje na úrovni konstruktorů konceptů. Na úrovni formulí potřebujeme pouze jednoduchou syntax na porovnávání konceptů či rolí a vyjádření vztahu individua ke konceptu či roli.

Pokud jsou a, b jména individuí, C a D koncepty, R a S role, pak

$C \sqsubseteq D$, $C \equiv D$, $R \sqsubseteq S$, $R \equiv S$, $C(a)$, $R(a, b)$, $\neg R(a, b)$, $\text{Disjoint}(R, S)$, $a \approx b$, a $a \not\approx b$ jsou atomické formule.

3.4 Sémantika deskripčních logik

3.4.1 Interpretace

Interpretace I deskripčních logik obsahuje neprázdnou nosnou množinu (množinu individuí) Δ^I a interpretační funkci, která přiřazuje každému prvku z množiny jmen individuí N_I prvek z nosné množiny, každému prvku z množiny jmen konceptů N_C podmnožinu nosné množiny a každému prvku z množiny jmen rolí N_R podmnožinu kartézského součinu nosné množiny se sebou samou. Formálně:

Pro každé $i \in N_I$: $i^I \in \Delta^I$, pro každé $C \in N_C$: $C^I \subseteq \Delta^I$, pro každé $R \in N_R$: $R^I \subseteq (\Delta^I \times \Delta^I)$.

3.4.2 Interpretace složených výrazů

Cílem interpretace každé logiky je získat pravdivostní hodnoty formulí. Toho dosáhneme induktivním dodefinováním interpretační funkce pro složené výrazy. Následující přehled obsahuje definice interpretací pro všechny konstruktory uvedené v sekci o sémantice deskripčních logik.

Nechť jsou a, b, a_1, \dots, a_n jména individuí, A atomický koncept, C, D koncepty, R role, pak pro konstruktory konceptů definujeme induktivně interpretaci následujícím způsobem:

$$\top^I = \Delta^I$$

$$\perp^I = \emptyset$$

$$\{a_1, \dots, a_n\}^I = \{a_1^I, \dots, a_n^I\}$$

$$(\neg A)^I = \Delta^I \setminus A^I$$

$$(\neg C)^I = \Delta^I \setminus C^I$$

$$(C \sqcup D)^I = C^I \cup D^I$$

$$(C \sqcap D)^I = C^I \cap D^I$$

$$(\exists R. \top)^I = \{a \in \Delta^I \mid (\exists b) (a, b) \in R^I\}$$

$$(\geq n R)^I = \{a \in \Delta^I \mid |\{b \in \Delta^I \mid (a, b) \in R^I\}| \geq n\}$$

$$(\leq n R)^I = \{a \in \Delta^I \mid |\{b \in \Delta^I \mid (a, b) \in R^I\}| \leq n\}$$

$$(\equiv n R)^I = \{a \in \Delta^I \mid |\{b \in \Delta^I \mid (a, b) \in R^I\}| = n\}$$

$$(\exists R. C)^I = \{a \in \Delta^I \mid (\exists b) (a, b) \in R^I \wedge b \in C^I\}$$

$$(\forall R. C)^I = \{a \in \Delta^I \mid (\forall b) (a, b) \in R^I \rightarrow b \in C^I\}$$

$$(\geq n R. C)^I = \{a \in \Delta^I \mid |\{b \in \Delta^I \mid (a, b) \in R^I \wedge b \in C^I\}| \geq n\}$$

$$(\leq n R. C)^I = \{a \in \Delta^I \mid |\{b \in \Delta^I \mid (a, b) \in R^I \wedge b \in C^I\}| \leq n\}$$

$$(\equiv n R. C)^I = \{a \in \Delta^I \mid |\{b \in \Delta^I \mid (a, b) \in R^I \wedge b \in C^I\}| = n\}$$

Nechť jsou R, S, U role a C koncept, pak pro konstruktory rolí definujeme induktivně interpretaci takto:

$$U^I = \{(a,b) \mid a \in \Delta^I \wedge b \in \Delta^I\}$$

$$(R^{-})^I = \{(a,b) \mid (b,a) \in R^I\}$$

$$(R^+)^I = \bigcup_{n \geq 1} (R^I)^n$$

$$(R^*)^I = \bigcup_{n \geq 0} (R^I)^n$$

$$(\neg R)^I = (\Delta^I \times \Delta^I) \setminus R^I$$

$$(R \sqcup S)^I = R^I \cup S^I$$

$$(R \sqcap S)^I = R^I \cap S^I$$

$$(R \circ S)^I = R^I \circ S^I$$

$$(id(C))^I = \{(a,a) \mid a \in C^I\}$$

$$(R|_C)^I = R^I \cap (\Delta^I \times C^I)$$

Nechť jsou a, b jména individuí, C, D koncepty, R, S role, pak pro atomické formule definujeme induktivně interpretaci následovně:

$$(C \sqsubseteq D)^I = (C^I \subseteq D^I)$$

$$(C \equiv D)^I = (C^I = D^I)$$

$$(R \sqsubseteq S)^I = (R^I \subseteq S^I)$$

$$(R \equiv S)^I = (R^I = S^I)$$

$$(C(a))^I = (a^I \in C^I)$$

$$(R(a,b))^I = ((a^I, b^I) \in R^I)$$

$$(\neg R(a,b))^I = ((a^I, b^I) \notin R^I)$$

$$\text{Disjoint}(R, S)^I = (R^I \cap S^I = \emptyset)$$

$$(a \approx b)^I = (a^I = b^I)$$

$$(a \not\approx b)^I = (a^I \neq b^I)$$

3.4.3 Teorie a modely teorií

Definice pojmů teorie a modelu teorie jsou v deskripčních logikách analogické k běžným definicím, nicméně jsou potřeba pro definice pojmů specifických pro deskripční logiky: pro pojmy subsumpce konceptů a pro pojem splnitelnosti konceptu, které jsou klíčové pro definování rozhodovacích problémů na deskripčních logikách, a proto zde uvedeme definice pro všechny tyto pojmy.

Teorie T deskripční logiky DL je množina formulí deskripční logiky DL.

Model teorie T je taková interpretace logiky DL, že všechny formule z T jsou splněné.

Konzistentní teorie T je taková teorie, která má model.

Splnitelný koncept C logiky DL vzhledem k teorii T je takový koncept, že existuje interpretace I, která je modelem T, taková, že C^I je neprázdná množina.

Koncept C je subsumpcí D vzhledem k teorii T, pokud pro každý I model T platí, že C^I je podmnožinou D^I .

Individuum a je instancí konceptu C (splňuje koncept C, spadá pod koncept C) vzhledem k teorii T, pokud pro každý model T platí, že interpretace individua a je prvkem interpretace konceptu C.

Rozhodování, zda teorie je konzistentní, zda v zadané teorii je koncept splnitelný, zda je jeden koncept subsumpcí druhého a zda individuum spadá pod daný koncept, jsou hlavní rozhodovací problémy deskripčních logik jako formalismu reprezentace znalostí.

3.5 Konkrétní deskripční logiky

Rodina deskripčních logik je velice početná a neustále rostoucí. Navíc je velmi mladá a neprozkoumaná, takže nemá dostatečně systematická a ustálená jména pro jednotlivé deskripční logiky. Jistý systém pojmenovávání přesto existuje a je používán a obecně přijímán, nicméně je také neustále rozšiřován a průběžně upravován spolu s tím, jak se rozšiřují znalosti o deskripčních logikách. Z tohoto důvodu nemůže být následující popis rodiny deskripčních logik a jejich pojmenovávání úplný. Částečný popis rodiny deskripčních logik nabízí Baader(2003), doplňkový popis nabízí Rudolph(2011), ale asi nejúplnější popis uvádí Horrocks(1997).

V rodině deskripčních logik se částečně z historických a částečně z technických důvodů považují některé deskripční logiky za základní a zbylé za rozšíření (nebo i omezení) základních logik o některé konstruktory konceptů nebo rolí nebo i o speciální požadavky (dodatečné podmínky) na sémantiku. Příkladem takové dodatečné podmínky může být

požadavek na tranzitivitu rolí nebo třeba požadavek, aby různá jména individuí nutně označovala různá individua.

Za základní jsou v rodině deskripčních logik považovány logiky *AL*, *FL* a *EL*.

3.5.1 Jazyk deskripční logiky *AL*

Deskripční logika *AL* (zkratka pochází z anglického attributive language) poprvé popsána v (Schmidt-Schauss, Smolka 1991) je nejčastěji používaná základní deskripční logika. Většinou je ovšem používána nikoliv samotná, ale pouze jako základní jazyk, který je obohacován o různá rozšíření a až po takovém rozšíření používán nebo zkoumán. Deskripční logika *AL* obsahuje tyto konstruktory konceptů: univerzální a prázdný koncept, koncept atomickou negací, koncept průnikem, koncept omezenou existenční kvantifikací a koncept obecnou kvantifikací, tj.: *pokud jsou a,b jména individuí, A je atomický koncept, C, D jsou koncepty a R je role, pak:*

$$\top, \perp, \neg A, C \sqcap D, \exists R.C, \forall R.C \text{ jsou koncepty v jazyce } AL$$

a

$$C \sqsubseteq D, C \equiv D, C(a), R(a,b) \text{ jsou atomické formule v jazyce } AL.$$

3.5.2 Jazyk deskripční logiky *FL*

Deskripční logika *FL* (*FL* znamená frame language a odkazuje k původu této logiky v systémech rámců) obsahuje konstruktory konceptů: koncept průnikem, koncept silnou existenční kvantifikací a koncept obecnou kvantifikací a konstruktor rolí restrikcí na koncept tj.: *pokud a,b jsou jména individuí, C, D jsou koncepty a R je role, pak:*

$$C \sqcap D, \exists R.C, \forall R.C \text{ jsou koncepty v jazyce } FL, R|_c \text{ je role jazyka } FL$$

a

$$C \sqsubseteq D, C \equiv D, C(a), R(a,b) \text{ jsou atomické formule v jazyce } FL.$$

Jazyk *FL* oslabený o konstruktor role restrikcí se nazývá FL^- , pokud tento dále oslabíme o omezenou existenční kvantifikaci, dostaneme jazyk v literatuře nazývaný FL^0 .

3.5.3 Jazyk deskripční logiky *EL*

Deskripční logika *EL* je ze základních deskripčních logik nejjednodušší, protože obsahuje pouze dva konstruktory konceptů, a to průnikem a silnou existenční kvantifikací, tj.:
pokud jsou a,b jména individuí, C, D jsou koncepty a R je role, pak:

$$C \sqcap D, \exists R.C \text{ jsou koncepty v jazyce } EL$$

a

$$C \sqsubseteq D, C \equiv D, C(a), R(a,b) \text{ jsou atomické formule v jazyce } EL.$$

Navzdory tomu, že jazyk *EL* je poměrně slabý v porovnání s ostatními deskripčními logikami (pro jazyky odvozené z jazyka *EL* se někdy používá pojem „lightweight description logics“), jsou dobře použitelné například pro modelování medicínských ontologií, jak ukazuje jejich použití projektem SNOMED-CT.

3.5.4 Rozšíření syntaxe základních jazyků deskripčních logik

Základní jazyky představené výše jsou poměrně jednoduché a chybí v nich mnoho konstrukcí a operací, na které jsme zvyklí z jiných logik. Tyto nedostatky se řeší rozšiřováním základních jazyků deskripčních logik. Tato rozšíření a jejich kombinace jsou příčinou početnosti, různorodosti a značné variability expresivní síly deskripčních logik. Převážná část těchto rozšíření spočívá pouze v přidání dalších konstruktorů konceptů.

Většina užívaných rozšíření syntaxe o další konstruktory konceptů má svá ustálená označení, což obvykle jsou jednotlivá písmena abecedy. Je zavedena konvence, že pokud máme jazyk, který chceme rozšířit o některé ze syntaktických rozšíření, budeme takto rozšířený jazyk značit jménem, které je prostým zřetězením označení původního jazyka a aplikovaného rozšíření. Tento postup je možno iterovat a lze tak vytvořit sice podrobná, popisná, výstižná, ale zároveň i krátká označení různých jazyků deskripčních logik. Krátká proto, že základní jazyk má dva znaky a většina rozšíření přidává jménu právě jeden znak. V praxi se drtivá většina označení deskripčních logik vejde do pěti znaků.

Syntaxi a sémantiku těchto konstruktorů tvořící rozšíření jsme popsali v předchozích kapitolách, nyní zbývá popsat jejich obvyklé značení:

C značí přidání konstruktoru konceptů doplňkem.

U značí přidání konstruktoru konceptů sjednocením.

O značí přidání konstruktoru konceptů výčtem prvků.

E značí přidání konstruktoru konceptů silnou existenci kvantifikací.

N značí přidání konstruktoru konceptů slabým omezením počtu.

Q značí přidání konstruktoru konceptů silným omezením počtu.

I značí přidání konstruktoru rolí inverzí role.

H značí přidání syntaxe axiomů pro inkluzi rolí.

R značí obvykle přidání syntaxe axiomů pro průnik rolí .

3.5.5 Rozšíření úpravou sémantiky deskripčních logik

Nejčastější způsob úpravy sémantiky deskripčních logik spočívá v přidání dalších podmínek na interpretaci rolí. I když se takové úpravy neprojeví na syntaktické struktuře jazyka, mohou být přesto velmi významné, protože mohou snížit časovou náročnost odvozovacích algoritmů u takto rozšířených jazyků. Z podmínek na interpretaci rolí zmíníme dvě nejvýznamnější. První požaduje, aby část rolí byla funkcionální, tj. aby některé role neměly interpretaci pouze relaci, ale relaci, která je zároveň i funkcí, a druhá požaduje, aby některé role měly za interpretaci relaci, která je navíc tranzitivní.

Přidané F v označení jazyka znamená, že z množiny jmen rolí N_R vyčleníme podmnožinu N_F , u jejíž členů očekáváme, že interpretační relace bude navíc funkcí.

Přidané R_+ v označení jazyka znamená, že z množiny jmen rolí N_R vyčleníme podmnožinu N_{R_+} , u jejíž členů očekáváme, že interpretační relace bude navíc tranzitivní.

3.5.6 Deskripční logika S

Speciální označení S se používá pro deskripční logiku ALC_{R_+} . Tato logika je základem většiny zkoumaných silnějších deskripčních logik. Význam je těmto logikám přisuzován proto, že jsou obecně považovány za vhodnou kompromisní volbu mezi expresivní silou jazyka a výpočtovou složitostí inferenčních úloh. Z tohoto důvodu jsou logiky této rodiny (například $SHOIN$ nebo $SROIQ$) používány jako teoretický základ pro jazyky webových sémantik z rodiny OWL (Hitzler 2009).

4. Odvozovací úlohy reprezentace znalostí a deskripční logiky

Deskripční logiky jsou používány jako jazyky pro reprezentaci znalostí, což znamená, že při návrhu systému, který má řešit zadanou úlohu, nevstupují do procesu návrhu hned na začátku. Prvním krokem návrhu je totiž analýza zadání problému, rozhodnutí, že řešení takového problému bude vyžadovat znalostní bázi a inferenční stroj nad znalostní bází, dále analýzu znalostí, které chceme do znalostí báze ukládat, analýzu odvozovacích úloh, které nad znalostmi potřebujeme provozovat a z toho vyplývající hledání vhodného jazyka, použitelného pro reprezentaci znalostí v znalostní bázi. Teprve v tento okamžik, pokud svými předpoklady a schopnostmi splňují požadavky návrhu, jsou deskripční logiky jedním z kandidátů na použitý jazyk reprezentace znalostí. Porozumění tomuto postupu je důležité pro pochopení přechodu od formální sémantiky deskripčních logik k odvozovacím úlohám, umožněným touto sémantikou.

Konkrétně z toho vyplývá, že pojem teorie z formální sémantiky logiky při používání logiky v inferenčních systémech odpovídá pojmu báze znalostí a vysvětluje, proč jsou tyto pojmy často v textech (striktně vzato nesprávně) zaměňovány. Rovněž jsou zaměňovány pojmy formule a axiomu, je to z toho důvodu, že formule uložené ve znalostní bázi jsou považovány za pravdivé, a tedy na ně lze nahlížet jako na axiomy, a v algoritmické praxi se žádné jiné formule než uložené ve znalostní bázi nevyskytují, tedy se v textech o deskripčních logikách často setkáme s použitím pojmu axiom na místě, kde by bylo záhodno hovořit o formuli. Tato hrubá práce s pojmy se vyskytuje v mnoha odborných textech o deskripčních logikách a odvozovacích úlohách v nich a může být matoucí a velmi zdržující, pokud nevíme o výše uvedených jazykových problémech a důvodech jejich vzniku a nepřekládáme si text do korektní podoby.

4.1 Odvozovací úlohy

Jelikož primární účel deskripčních logik je taxonomická klasifikace pojmů a ukládání znalostí, odvozovací úlohy, které požadujeme po deskripčních logikách, odpovídají úlohám, které chceme být schopni provádět nad databází znalostí. U definic modelů a teorií v oddíle 3.4.3 je uvedeno několik pojmů, které slouží jako teoretický základ pro odvozovací úlohy u deskripčních logik. Zde následuje popis vztahů mezi pojmy z deskripčních logik a úlohami, které mají být proveditelné nad znalostní bází.

Z toho, že znalostní báze je v podstatě logická teorie, vyplývá, že řešení mnoha úloh reprezentace znalostí je v deskripčních logikách snadno uchopitelné pomocí standardních algoritmů. Problém konzistence znalostní báze odpovídá bezespornosti logické teorie, protože znalostní báze je logickou teorií deskripční logiky. Nalezení modelu znalostní báze je nalezení modelu teorie. Ověřování subsumpce se provádí vzhledem k znalostní bázi a ověřování, zda individuum splňuje koncept, se také provádí vzhledem k znalostní bázi.

4.2 Vzájemná převoditelnost odvozovacích úloh

Jelikož u deskripčních logik chceme být schopni řešit velké množství různých úloh, je užitečné, že mnoho z těchto úloh lze triviálně převést na jednu, na kterou potom stačí vyvinout jeden efektivní algoritmus.

U silnějších logik, které obsahují všechny booleovské operátory, lze pomocí negace převést problém subsumpce konceptů na problém splnitelnosti konceptu:

$$C \sqsubseteq D \text{ iff } (C \sqcap \neg D)^I = \emptyset$$

Dále lze podobným způsobem převést na problém konzistence báze znalostí všechny následující problémy: problém subsumpce konceptů, problém ekvivalentnosti konceptů, problém splnitelnosti konceptu, problémy, zda individuum náleží do konceptu nebo role.

4.3 Algoritmus strukturální subsumpce

Popíšeme algoritmus pro ověřování subsumpce použitelný pro slabší deskripční logiky. Tento algoritmus se poprvé objevuje v Levesque, Brachman (1987). Algoritmus je ukázán pro logiku FL_0 , ale lze rozšířit i pro další o málo silnější deskripční logiky. Pro logiku FL_0 má algoritmus dokonce polynomiální časovou složitost.

Algoritmus pracuje ve dvou fázích, v první fázi převádí vstupní koncepty do normální formy, kde *pokud* A_1, \dots, A_m jsou různé atomické koncepty, R_1, \dots, R_m jsou role a $C_1 \dots C_n$ jsou koncepty v normální formě, tak normální forma vypadá takto:

$$A_1 \sqcap \dots \sqcap A_m \sqcap \forall R_1.C_1 \sqcap \dots \sqcap \forall R_n.C_n$$

Po převedení do normální formy algoritmus přímočaře využívá následujícího tvrzení: *Pokud koncepty C, D jsou popořadě tvarů:*

$$A_1 \sqcap \dots \sqcap A_m \sqcap \forall R_1.C_1 \sqcap \dots \sqcap \forall R_n.C_n$$

$$B_1 \sqcap \dots \sqcap B_p \sqcap \forall S_1.D_1 \sqcap \dots \sqcap \forall S_r.D_r$$

tak: $C \sqsubseteq D$ právě tehdy, když platí obojí následující:

$$(\forall i)(1 \leq i \leq p)(\exists j)(1 \leq j \leq m) \wedge (B_i = A_j)$$

$$(\forall i)(1 \leq i \leq r)(\exists j)(1 \leq j \leq n) \wedge (S_i = R_j) \wedge (C_j \sqsubseteq D_i)$$

Algoritmus tedy ověřuje postupně tuto podmínku pro všechny podkoncepty konceptu C. Důkaz správnosti uvádí Brachman(2004) a některá rozšíření algoritmu na silnější logiky popisuje například Baader(2003).

4.4 Algoritmus sémantických tableaux

Algoritmus sémantických tableaux řeší problém splnitelnosti konceptu, ale je používán i na řešení problému subsumpce díky převoditelnosti problému subsumpce na problém splnitelnosti konceptu.

Algoritmus sémantických tableaux má jako svoji první fázi normalizaci vstupního konceptu C. Algoritmus potřebuje koncept ve tvaru NNF (negation normal form). Koncept je ve tvaru NNF, pokud se negace vyskytují pouze před jmény konceptů a nikoliv před koncepty složenými konstruktory konceptů. U silnějších deskripčních logik lze pro normalizaci použít de Morganových zákonů a pravidel pro kvantifikátory. Tedy v první fázi algoritmus převede vstupní koncept C na ekvivalentní koncept D. Pro tento koncept D pak algoritmus v druhé fázi konstruuje model. Protože algoritmus využívá backtracking, je pro ukládání modelu využit strom, kde ve vrcholech stromu jsou uloženy jednotlivé doposud získané informace o sestavovaném modelu. Pro dopředný krok v backtrackingu se vytváří nový potomek aktuálního vrcholu a ukládají se informace získané v daném dopředném kroku. Pro zpětný krok se aktuální vrchol včetně informací odebírá. Informace uložené ve vrcholech stromu jsou podmínky pro sestavovaný model. V počátečním vrcholu se ukládá podmínka, že koncept D má mít v modelu I individuum x , které jej bude splňovat, tedy:

$$x \in D^I$$

V dopředném kroku se podmínky zkopírují do nového vrcholu a pak se upravují na jednodušší podle následující sady produkčních pravidel:

Pokud a je individuum, C, D koncepty a vrchol obsahuje podmínku $a \in (C \sqcap D)^I$ a neobsahuje obě podmínky $a \in C^I$ a $a \in D^I$, tak přidej chybějící podmínky $a \in C^I$ a $a \in D^I$.

Pokud a je individuum, C koncept a R role a vrchol obsahuje podmínku $a \in (C \sqcup D)^I$ a neobsahuje alespoň jednu z podmínek $a \in C^I$ a $a \in D^I$, tak nedeterministicky vyber a přidej jednu z podmínek $a \in C^I$ nebo $a \in D^I$.

Pokud a je individuum, C, D koncepty a vrchol obsahuje podmínku $a \in (\forall R.C)^I$ a podmínku $(a, b) \in R^I$, pak přidej podmínku $b \in C^I$.

Pokud a je individuum, C, D koncepty a vrchol obsahuje podmínku $a \in (\exists R.C)^I$ a neobsahuje žádné z takové, že $z \in C^I$ a zároveň $(a, z) \in R^I$, pak přidej individuum z a podmínky $(a, z) \in R^I$ a $z \in C^I$.

Jedno z pravidel obsahuje nedeterministický výběr a právě ono je potenciálním zdrojem backtrackingu. Zpětný krok je nutný, pokud v podmínkách na konstruovaný model dojde ke sporu. Spor v podmínkách na konstruovaný model nastává, pokud pro nějaké individuum x z modelu máme podmínku, že musí být prvkem prázdného konceptu, nebo pokud individuum x musí být zároveň prvkem konceptu i jeho negace. Formálně:

$$(x \in \perp^I) \vee ((x \in C^I) \wedge (x \notin C^I)).$$

Algoritmus končí, pokud nelze aplikovat žádná pravidla a nedošlo ke sporu nebo pokud backtracking neumožňuje další procházení. Pokud algoritmus končí s tím, že není možno aplikovat žádná pravidla a nedošlo ke sporu, tak výsledná množina podmínek popisuje model splňující vstupní koncept D . V případě, že v každé větvi backtrackingu došlo ke sporu, není vstupní koncept splnitelný.

Uvedená množina produkčních pravidel je určena pro logiku ALC , při používání algoritmu sémantických tableaux na různá rozšíření logiky ALC je třeba rozšířit i produkční pravidla. Velkou výhodou algoritmu sémantických tableaux je, že toto je jediná změna, kterou je potřeba učinit při rozšíření logiky, nad kterou má algoritmus pracovat. Pro logiku ALC je algoritmus sémantických tableaux PSPACE těžký (Schmidt-Schauss, Smolka 1991). Popis rozšířeného algoritmu pro logiku $ALCN$ uvádí Baader(2003), stejně jako důkaz korektnosti a úplnosti.

4.5 Složitost subsumpce ve vybraných deskripčních logikách

Složitost odvozovacích úloh je spolu s expresivní silou hlavním kritériem pro posuzování deskripčních logik, a tedy jako součást hlavního kritéria je v každé práci o deskripčních logikách jedním ze dvou centrálních témat. V této kapitole uvádíme složitost odvozování pro vybrané logiky a odkazujeme na vybrané konkrétní práce, kde jsou tyto výsledky zdůvodňovány. Dále předkládáme seznam zajímavých logik, u kterých je složitost subsumpce známa.

Prvního významného výsledku dosáhli Levesque, Brachman (1987), kdy ukázali, že i tak malá změna, jako je změna z FL^- na FL , může způsobit tak zásadní skok ve složitosti subsumpce, jakým je v tomto případě skok z třídy složitosti PTIME do třídy složitosti CO-NP-těžkých úloh.

Obdobného, podobně významného výsledku dosáhli Schmidt-Schauss, Smolka (1991) s deskripční logikou AL , která z hlediska subsumpce patří do třídy PTIME, a deskripční logikou ALC , která je z hlediska subsumpce PSPACE těžká.

Systematický přístup přináší práce Donini et al. (1997), kde jsou studovány všechny kombinace jazyků $AL [U] [E] [N] [R]$, a ty, které ještě nebyly z hlediska složitosti klasifikovány, řadí do tříd složitosti dle složitosti odvozování. U ALN ukazuje, že patří do třídy PTIME, zatímco ALR , $ALER$ jsou NP-úplné, ALU CO-NP-úplná, $ALNR$, $ALUR$, $ALCNR$ jsou PSPACE těžké. Tento článek přináší také první kategorizaci deskripčních logik podle příčin složitosti odvozování. Spojuje poznatky z Schmidt-Schauss, Smolka (1991) a z Donini et al. (1992), uvádí a ukazuje, že v této uvažované skupině ($AL [U] [E] [N] [R]$) deskripčních logik jsou dva různé konstruktory příčinami růstu složitosti odvozování. Prvním konstruktorem, který zvyšuje složitost, je konstruktor sjednocením, který do algoritmů subsumpce zavádí nedeterminismus a jeho přítomnost způsobuje zvýšení složitosti do třídy NP-úplných úloh. Druhým konstruktorem, který zvyšuje složitost subsumpce, je konstruktor silné existenční kvantifikace, který splnitelnost činí CO-NP-těžkou. A konečně, pokud se oba tyto zdroje výpočetní složitosti setkají v jednom jazyce, je výsledný jazyk ve třídě PSPACE těžkých.

U následujících logik je známo, že u nich má subsumpce pouze polynomiální složitost:

FL^- , FL^-R , AL , ALN , AL se skládním rolí, AL s inverzí rolí.

U následujících logik je známo, že u nich je subsumpce NP-úplná:

$FL-E$, ALE , $ALER$.

U následujících logik je známo, že u nich je subsumpce CO-NP-těžká:

*ALU, ALN s inverzí rolí, FL⁻R s inverzí rolí, FL⁻R se skládáním rolí,
FL⁻ se skládáním a inverzí rolí.*

U následujících logik je známo, že u nich je subsumpce PSPACE těžká:

FL, ALC, ALNE, ALNR, ALUR, ALCNR.

Závěr

V tomto textu byly prezentovány deskripční logiky jako moderní nástroj reprezentace znalostí v rámci umělé inteligence i jako samostatný jazyk formální logiky. Jelikož se jedná o stále živě se rozvíjející obor vědeckého zkoumání, lze předpokládat, že některá zde prezentované výsledky vědeckých výzkumů deskripčních logik časem pozbudou platnosti. Nicméně zde prezentované obecné koncepty a fundamentální ideje deskripčních logik tímto ohrožené nejsou.

Deskripční logiky momentálně slibují dva potenciálně značné přínosy. Pro informatiku obecně se mohou stát univerzálním jazykem na ukládání a zpracovávání informací, a tedy znalostí v síťových systémech. Univerzální jazyk na ukládání znalostí by mohl zpřístupnit jednomu programu data, která v současnosti jsou reprezentována v různých formátech a přístupná pomocí různých dotazovacích jazyků, což minimálně hodně znesnadňuje a často i úplně znemožňuje jejich společné využití.

Druhým z těchto potenciálních velmi významných přínosů by mohlo být porozumění příčinám rozdílů ve výpočtové složitosti u různých jazyků. Tím, že deskripční logiky představují tak široké spektrum jazyků z hlediska jejich expresivní síly a výpočtové složitosti odvozovacích úloh a zároveň jsou velice jednotné až chudé z hlediska rozmanitosti syntaktické a sémantické struktury, poskytují zajímavý zjednodušený prostor pro zkoumání příčin těchto rozdílů. Tyto rozdíly v expresivitě a jejich vliv na výpočtovou složitost inference u deskripčních logik jsou stále intenzivně zkoumány, některé výsledky jsou v tomto textu prezentovány a je možné, zrovna na poli deskripčních logik přesáhne množství znalostí o tomto tématu onu kritickou mez, že bude možné stanovit nějaké obecněji platné zákonitosti pro tuto problematiku.

Seznam použité literatury

BAADER, Franz, et al. 2003. The Description Logic Handbook. Cambridge University Press. ISBN 978-0521781763

BRACHMAN, Ronald J., LEVESQUE, Hector J. 2004. Knowledge Representation and Reasoning. Morgan Kaufmann. ISBN 978-1558609327

BRACHMAN, Ronald J., SCHMOLTZE, James G. 1985. An Overview of the KL-ONE Knowledge Representation System in Cognitive science 9: 171-216

DONINI, Francesco M., et al. 1997. The Complexity of Concept Languages in Information and Computation 134: 1-58

HARMELEN, Frank van, et al. 2008. Handbook of Knowledge Representation. Amsterdam: Elsevier. ISBN 978-0-444-52211-5

HITZLER, Pascal, et al. Edts. 2009. OWL 2 Web Ontology Language: Primer. <http://www.w3.org/TR/owl2-primer/>

HORROCKS, Ian R. 1997. Optimising Tableaux Decision Procedures for Description Logics

KRÖTZSCH, Marcus, SIMANČÍK, František, HORROCKS Ian. 2013. A Description Logic Primer

LEDERBERG, Joshua. 1987. How DENDRAL was conceived and born. HMI 1987 Proceedings of ACM conference on History of medical informatics.

LEVESQUE, Hector J., BRACHMAN, Ronald J. 1987. Expressiveness and tractability in knowledge representation and reasoning

LINDSAY, Robert K., BUCHANAN, Bruce G. FEIGENBAUM, Edward A. and LEDERBERG, Joshua. 1980 . Applications of Artificial Intelligence for Organic Chemistry: The Dendral Project. McGraw-Hill Book Company.

LEDERBERG, Joshua. 1987. How DENDRAL was conceived and born. ACM Conference on the History of Medical Informatics

MINSKY, Marvin. 1974. A Framework for Representing Knowledge. MIT-AI Laboratory Memo, Reprinted in The Psychology of Computer Vision

NEWELL, Allen, SIMON, Herbert A. 1956. The Logic Theory Machine a Complex Information Processing System. Rand corporation manual

NEWELL, Allen, SIMON, Herbert A. 1976. Computer science as empirical inquiry: symbols and search

RUDOLPH, Sebastian. 2011. Foundations of Description Logics. Karlsruhe Institute of Technology

RUSSELL, Stuart Jonathan, NORVIG, Peter. 2003. Artificial Intelligence: A Modern Approach 3rd edition, Upper Saddle River: Prentice Hall. ISBN 978-0136042594

SCHMIDT-SCHAUSS, Manfred, SMOLKA, Gert. 1991. Attributive concept descriptions with complements in Artificial Intelligence 48: 1-26 Amsterdam: Elsevier.