Vážený/á studující předmětu Reprezentace znalostí

v předcházejících semestrech jste se v předmětech Logické základy umělé inteligence 1 a 2 seznámil/a s logikou prvního řádu neboli predikátovou logikou, zahrnující jako svoji podoblast i logiku výrokovou, která patří k základnímu teoretickému vybavení informatika. Zvykl/a jste si na určitý způsob zjednodušování modelovaného světa, který je při formální reprezentaci znalostí nutný. Studium předmětu Reprezentace znalostí poněkud zpochybní Vaše přesvědčení, že prostředky jazyka logiky prvního řádu jsou pro reprezentaci skutečných znalostí dostačující. Setkáte se zde s řadou přístupů, vycházejících z logiky prvního řádu, které mají za cíl vyšší stupeň přiblížení k manipulaci se znalostmi, tak jak se provádějí v lidských mozcích. Seznámíte se též s algebraickými prostředky reprezentace vztahů mezi základními objekty lidské mysli, kterými jsou pojmy. Seznámíte se s prostředky vytváření hierarchických struktur pojmů – formálních ontologií, umožňujících strukturovat informace v rámci „sémantického webu“.

Předmět je rozdělen do 12 lekcí - témat odpovídajících v podstatě náplni přednášek prezenčního studia předmětu. Studium každé z nich by Vám mělo zabrat přibližně 1.5 hodiny.

Doporučuji Vám studovat text po jednotlivých lekcích, odpovídat na kontrolní otázky a řešit úkoly zadané v textu.

Přeji Vám dostatek píle a trpělivosti při studiu předmětu.

Alena Lukasová

1 REPREZENTACE ZNALOSTÍ V ASOCIATIVNÍCH SÍTÍCH

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se seznámíte s názorným grafickým způsobem reprezentace znalostí, který vychází ze zjednodušené verze predikátové, resp. klauzulární logiky.Uvidíte, že asociativní sítě jsou nejen vhodným prostředkem pro konceptuální modelování při vytváření znalostních bází, ale že v nich lze vyjádřit i vzájemné souvislost reprezentovaných znalostí, na jejichž základě je (viz následující lekci) možno dokazovat a odvozovat logické důsledky vyplývající ze znalostních bází.

1.1 ASOCIATIVNÍ SÍŤ

Asociativní (sémantické) sítě zavedl v roce 1968 M.R.Quillian pro účel modelování sémantiky anglických vět. Stejně jako v predikátové logice vychází reprezentace asociativními sítěmi z atomů vhodně zvolených predikátů. Na rozdíl od logiky 1. řádu však asociativní síť disponuje pouze dvoumístnými predikáty, pouze takové relace lze totiž reprezentovat jako síť, která má své hrany a uzly. Hrany nesou jako svá návěští predikátové symboly, uzly symboly přípustné v predikátových atributech, tj. termy pojmenovávající objekty reprezentovaného světa - *referenčního systému*.

Asociativní sítě nedisponují prostředky pro reprezentaci univerzální a existenční kvantifikace. Formule logiky prvního řádu je tedy pro možnost reprezentace sítí nutno upravit do speciálního klauzulárního tvaru. To je problém, který asociativní sítě sdílejí s klauzulární logikou. Jeví se proto vhodné, vypořádat se s problémem reprezentace kvantifikátory vázaných proměnných obdobně jako je tomu v případě klauzulární logiky: Všechny proměnné, označované zde symboly začínajícími velkým písmenem, jsou univerzálně vázány, existenčně vázané proměnné lze skolemizací upravit na existenční termy.

Asociativní síť je jedním z prostředků reprezentace znalostí, schopných zohlednit jejich strukturovanost, neboť na rozdíl od např. klauzulární logiky jsou v nich reprezentovány jednotlivé části znalosti sdruženy do významových celků.

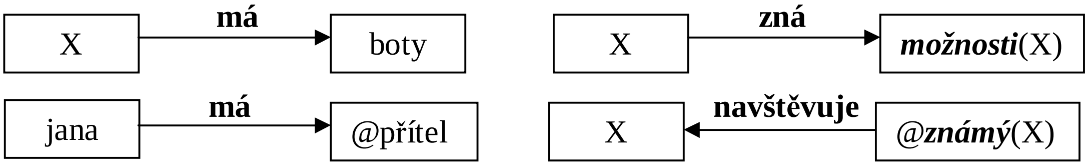
Příklad 1.1

Atomické výroky jako „Děda hledá brýle“, „Babička jezdí na kole“, „Každý má boty.“, „Každý zná své možnosti.“, „Jana má nějakého přítele.“, „Každého navštěvuje někdo známý.“, jejichž reprezentace v klauzulární logice by mohla mít tvar

hledá(děda, brýle), **jezdí_na**(babička, kolo), **má**(X, boty), **zná**(X, ***možnosti***(X)),

má(jana, @přítel), **navštěvuje**(@***známý***(X), X),

jsou zde reprezentovatelné elementy sítě – jejími *vektory* :



obr. 1.1

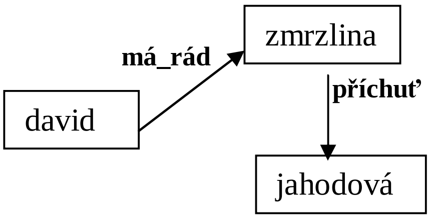
Definice 1.1

Asociativní síť je ohodnocený graf sestávající z uzlů, ohodnocených termy, a hran, ohodnocených binárními predikátovými symboly, přičemž hrany spojují některé dvojice uzlů.

Způsobem, který byl až doposud uvažován, lze reprezentovat asociativními sítěmi pouze malou skupinu tvrzení, neboť ty jsou především určeny pro reprezentaci binárních relací. Pro vícemístné relace je třeba je upravit do složitějších struktur.

Příklad 1.2

Tvrzení "David má rád jahodovou zmrzlinu.", které je možno vyjádřit ternárním predikátem, jehož schéma je **má_rád**(<kdo>, <co>, <jaký>), lze reprezentovat sítí :

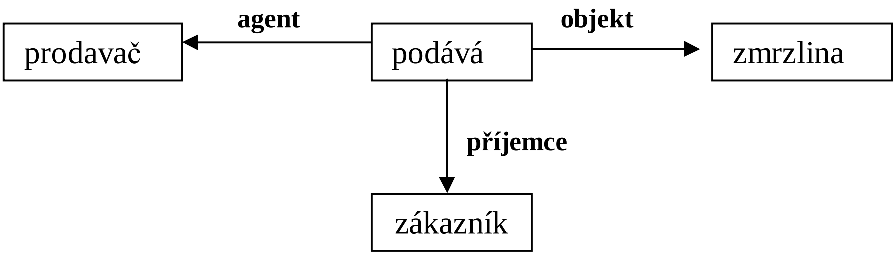


obr. 1.2

Mnoho sloves, která jsou zpravidla reprezentována predikáty, je vázáno na přímý a současně na nepřímý předmět.

Příklad 1.3

„Prodavač podává zákazníkovi zmrzlinu.“

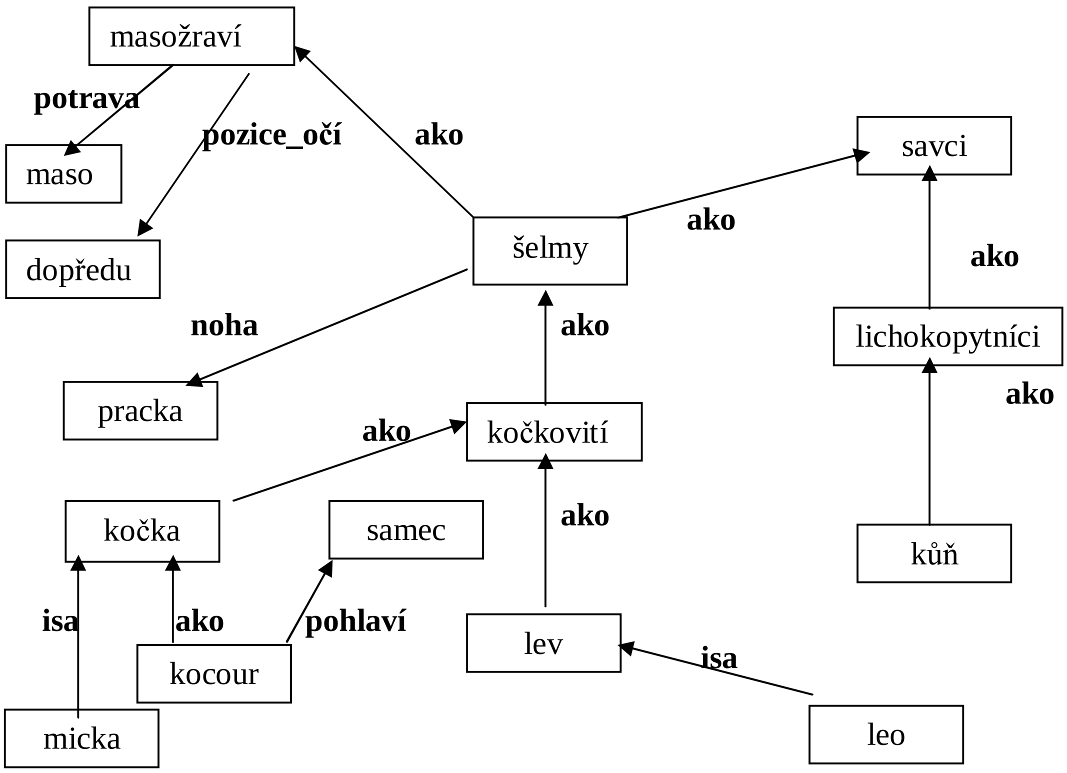


obr. 1.3

V asociativních sítích se často používají hrany označené predikátovými symboly **agent**, **objekt**, **nástroj**, …, resp. predikáty schopné reprezentovat množinovou inkluzi a hierarchii **:** **ako** - "A kind of" **isa** - "B is A" :

Příklad 1.4

Asociativní síť živočišných druhů a jejich vlastností



obr. 1.4

1.2 JAZYK A ZNALOSTNÍ BÁZE ASOCIATIVNÍCH SÍTÍ

Aby systém asociativních sítí byl formálním (axiómatickým) systémem, je třeba

1. definovat syntax jeho jazyka
2. stanovit, jakým způsobem probíhá formální odvozování v asociativní síti, tj
   - definovat, co je výchozí znalostní báze, sestávající ze speciálních axiómů teorie
   - charakterizovat odvozovací pravidla, jimiž se ze znalostní báze generují věty teorie
3. definovat jeho sémantiku a dokázat jeho sémantickou korektnost

1.2.1 Jazyk asociativních sítí

Definice 1.2

Jazyk asociativních sítí disponuje těmito typy symbolů :

1.a) grafické symboly označující uzly sítě <návěští

b) termy v návěštích uzlů - znakové řetězce

- pro označení proměnných (začínající velkým písmenem) : X,Y,..
- pro označení konstant (začínající malým písmenem nebo číslicí) : dům, kočka, 12,…
- pro funkční symboly : ***sin***(X), ***matka***(X),...
- pro existenční termy (začínající symbolem @) : @někdo, @známý(X),...

2.a) grafické symboly označující hrany sítě $\xrightarrow{\text{<návěští hrany>}}$

b) binární predikátové symboly v návěštích hran : **ako**(X,Y), **bydlí**(X,Y),…
(atributy v závorkách se v grafu neuvádějí).

Definice 1.3

Atom asociativní sítě tvoří *vektor* sestávající ze dvou uzlů, jejichž návěštími jsou termy, a jedné spojující hrany, jejímž návěštím je predikátový symbol.

$$\boxed{\text{< term 1 >}} \xrightarrow{\text{< predikátový symbol >}} \boxed{\text{< term 2 >}}$$

Asociativní síť tvoří množina (ne nutně) vzájemně propojených atomických vektorů.

1.2.2 Znalostní báze

V rámci znalostních bází asociativních sítí jsou tvrzení reprezentována sítěmi těchto typů :

- *nepodmíněné sítě* - *bázové* (fakta)
  - *univerzální*
- podmínky - bázové
  - univerzální

Příklad 1.4 představuje bázovou nepodmíněnou síť, obsahující v návěštích svých uzlů výhradně konstanty.

Příkladem *univerzální nepodmíněné sítě* reprezentující tvrzení „Každý hledá své štěstí.“ je vektor

$$\boxed{\text{X}} \xrightarrow{\textbf{hledá}} \boxed{\textbf{\textit{štěstí}}\text{(X)}}$$

Podmínky v asociativních sítích představují, podobně jako je tomu v predikátové nebo klauzulární logice, pravidla s antecedentem a konsekventem. Na skutečnosti, zdali se v návěštích jejich uzlů vyskytují nebo nevyskytují proměnné, závisí jejich zařazení mezi *univerzální* nebo *bázové podmínky*. Obecný tvar podmínky je v asociativních sítích

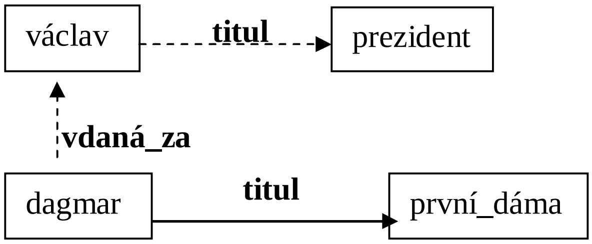
$$\mathbf{q} \text{ if } \mathbf{p}_1, \mathbf{p}_2 ,...\mathbf{p}_n$$

Podmínky v sítích mohou mít více než jeden atom v antecedentu, ale <u>pouze jeden</u> (na rozdíl od klauzulární logiky) v konsekventu.

Podmínky se zakreslují tak, že se atomy antecendentu označují čárkovanými hranami, konsekventu plnou čarou.

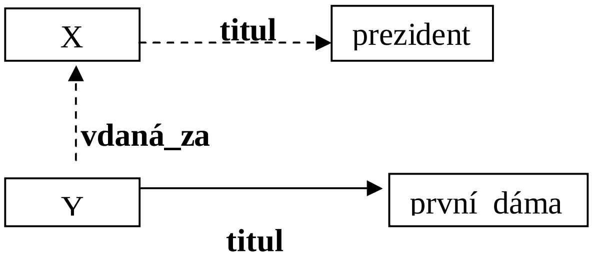
Příklad 1.5

Bázová podmínka reprezentující tvrzení „Dagmar je první dáma, jestliže se vdala za prezidenta Václava.“ :



obr. 1.5

Náhradou některých konstantních návěští uzlů proměnnými se z dané bázové podmínky na obr. 1.5 stane *podmínka univerzální*.



obr. 1.6

Nepodmíněné sítě reprezentují fakta.

Podmínky vyjadřují, co z čeho vyplývá.

Úkol 1.1

a) Vytvořte asociativní síť příbuzenských vztahů ve Vaší rodině,

b) Zavedením proměnných tuto síť zobecněte

c) Vytvořte podmínku, která vyjadřuje stav být babičkou (dědečkem)

Asociativní síť je ohodnocený graf sestávající z uzlů, ohodnocených termy, a hran, ohodnocených binárními predikátovými symboly, přičemž hrany spojují některé dvojice uzlů.

Jazyk asociativních sítí disponuje těmito typy symbolů :

1.a) grafické symboly označující *uzly sítě*

b) *termy* v návěštích uzlů - znakové řetězce

pro označení proměnných (začínající velkým písmenem) : X,Y,..

pro označení konstant (začínající malým písmenem nebo číslicí) : dům, kočka, 12,…

pro funkční symboly : ***sin***(X), ***matka***(X),...

pro existenční termy (začínající symbolem @) : @někdo, @známý(X),...

2.a) grafické symboly označující *hrany sítě*

b) binární predikátové symboly v návěštích hran : **ako**(X,Y), **bydlí**(X,Y),…

Atom asociativní sítě tvoří *vektor* sestávající ze dvou uzlů, jejichž návěštími jsou termy, a jedné spojující hrany, jejímž návěštím je predikátový symbol.

Asociativní síť tvoří množina (ne nutně) vzájemně propojených atomických vektorů.

V rámci znalostních bází asociativních sítí jsou tvrzení reprezentována sítěmi těchto typů :

nepodmíněné sítě - bázové (fakta)
- univerzální

podmínky - bázové
- univerzální

Podmínky v asociativních sítích představují, podobně jako je tomu v predikátové nebo klauzulární logice, pravidla s antecedentem a konsekventem. Na skutečnosti, zdali se v návěštích jejich uzlů nevyskytují nebo vyskytují proměnné, závisí jejich zařazení mezi *univerzální* nebo *bázové nepodmíněné sítě/podmínky*.

Kontrolní otázky :

1. Definujte syntax jazyka asociativních sítí.
2. Jaký je rozdíl mezi nepodmíněnou sítí a podmínkou ?
3. Čím se liší bázová síť od univerzální sítě ?

2 ODVOZOVÁNÍ V ASOCIATIVNÍCH SÍTÍCH

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se dozvíte, že i v případě reprezentace znalostí názorným grafickým způsobem je možno dokazovat věty a odvozovat logické důsledky vyplývající ze znalostních bází.

2.1 PRINCIP ODVOZOVÁNÍ ZE ZNALOSTNÍ BÁZE

Definice 2.1 (pravidel odvozování)

Odvozováním v asociativních sítích se rozumí doplňování dalších vektorů nebo termů v návěštích jejich uzlů do sítě, která je pro toto odvozování hlavní sítí, pomocí jiné zvláštní sítě některým z těchto odvozovacích pravidel :

Pravidlo uniformní substituce :

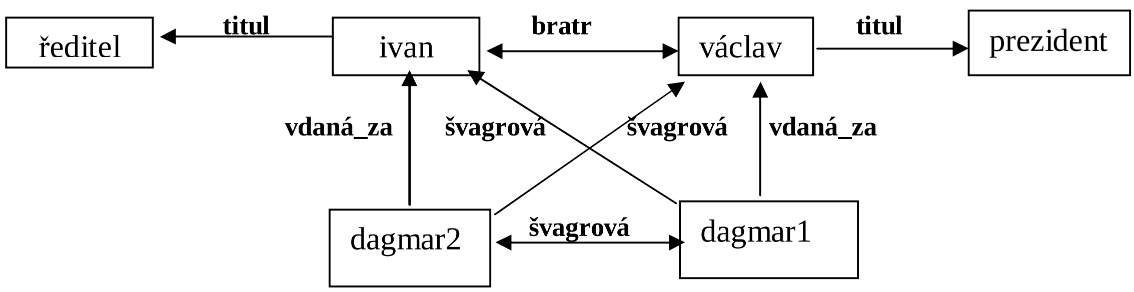
Jestliže se všechny ohodnocené hrany zvláštní sítě objeví v hlavní síti, obsahující proměnné jako návěští jimi spojených uzlů, pak lze návěští z uzlů zvláštní sítě substituovat uniformně za návěští odpovídajících uzlů sítě hlavní, označených proměnnými.

Pravidlo transferu :

Jestliže se všechny čárkované hrany antecedentu zvláštní sítě objeví (jako plné) v hlavní síti, přičemž návěští odpovídajících si uzlů se buď shodují (nebo jsou v některé ze sítí proměnnými), pak lze (event. po aplikaci pravidla uniformní substituce) plný vektor konsekventu ze zvláštní sítě doplnit do sítě hlavní.

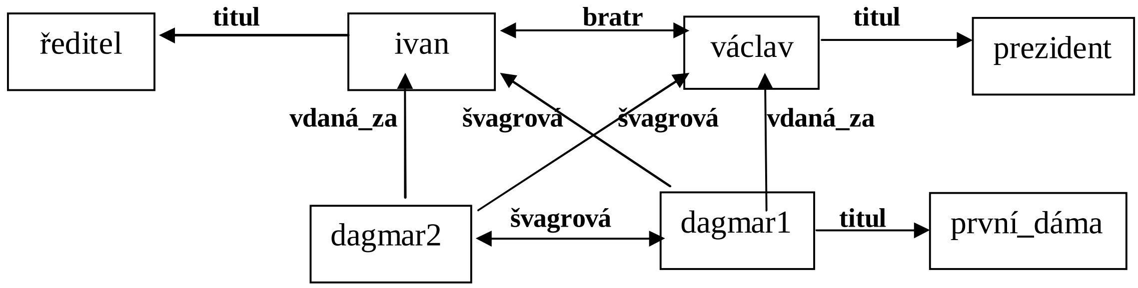
Příklad 2.1

Hlavní síť reprezentující některé rodinné vztahy v rodině prezidenta Havla :



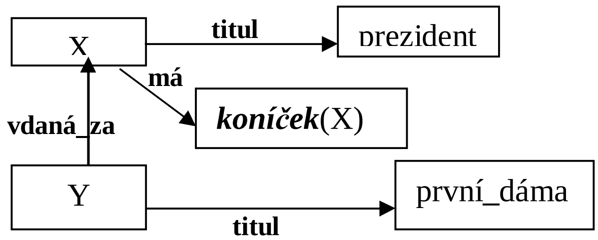
obr. 2.1

Doplněním vektoru konsekventu univerzální podmínky z příkladu 2.1 (zvláštní sítě) do hlavní sítě podle pravidla transferu vznikne síť :



obr. 2.2

Je-li hlavní sítí univerzální síť, která mimo jiné tvrdí, že i osoba v prezidentské funkci má svého koníčka,

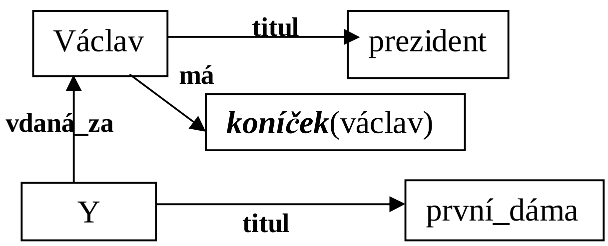


obr. 2.3

a zvláštní sítí vektor

obr. 2.4

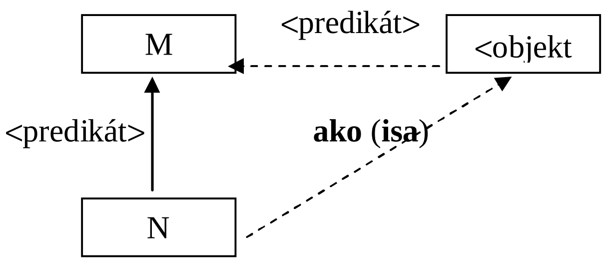
je možno podle pravidla uniformní substituce do této hlavní sítě doplnit za X term václav :



obr. 2.5

a získat tak novou univerzální síť, která reprezentuje tvrzení, že kterákoliv osoba vdaná za prezidenta Václava je první dámou. Požadavek uniformnosti substituce zaručí přiřazení každému v prezidentské funkci jeho vlastního jediného koníčka.

Asociativní síti obsahující **ako**(**isa**)-hrany (viz příklad 1) a určitý predikát na zbývající hraně antecedentu lze doplnit do konsekventu týž predikát podle obecného pravidla :



obr. 2.6

Např. při doplnění informace, že kočičí noha je pracka, do sítě příkladu 1.4 jde o doplnění hrany **noha** s hlavou (šipkou) u uzlu s návěštím pracka a koncem u uzlu s návěštím kočka.

Při odvozování v asociativních sítích doplňujeme další vektory nebo termy v návěštích uzlů do hlavní sítě pomocí jiné zvláštní sítě, a to některým z odvozovacích pravidel *uniformní substituce* nebo *transferu.*

2.2 SÉMANTIKA ASOCIATIVNÍCH SÍTÍ

2.2.1 Interpretace asociativní sítě a její struktura

Pro přiřazení struktury dané síti pro možnost její interpretace platí podobně jako v klauzulární logice určitá pravidla.

Definice 2.2

Strukturu interpretace asociativní sítě tvoří trojice množin {**W**, **F**, **R**}, kde

(1) **W** je množina objektů světa, který má být asociativní sítí reprezentován.

(2) Každé konstantě sítě je přiřazen nějaký objekt universa diskursu **W** daného světa - její denotát.

(3) Každé existenční konstantě sítě je přiřazeno jako množina možných denotátů universum diskursu **W**.

(4) Každé funkci je přiřazen jako její denotát předpis z množiny **F** pro její vyhodnocení.

(5) Každému n-árnímu existenčnímu funktoru sítě je přiřazena jako množina možných denotátů množina všech n-árních funkcí nad universem diskursu **W**.

(6) Každému predikátu sítě je přiřazen jeho denotát - relace z množiny **R** definovaná na daném universu diskursu **W**.

2.2.2 Pravdivost bázových vektorů a sítí

Definice 2.3

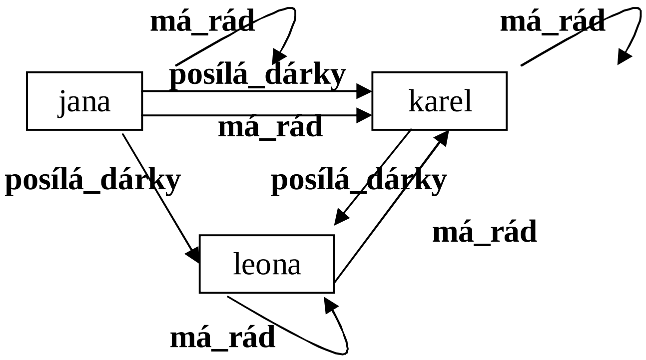
Bázový vektor $\boxed{A} \xrightarrow{r} \boxed{B}$ je pravdivý ve struktuře **S** dané interpretace, jestliže denotát objektu A a denotát objektu B jsou v relaci R, která je denotátem predikátu **r** ve struktuře **S**. V opačném případě je nepravdivý.

Bázová nepodmíněná síť je pravdivá ve struktuře **S** dané interpretace, jsou-li v ní pravdivé všechny její bázové vektory.

Bázová podmínka je nepravdivá ve struktuře **S**, jestliže všechny její vektory antecedentu jsou pravdivé a vektor konsekventu je nepravdivý v **S**. Jinak je tato podmínka pravdivá.

Příklad 2.2

Svět reprezentovaný sítí se týká tří lidí : Jany, Karla a Leony. V tomto světě Jana posílá dárky Karlovi a Leoně, Karel posílá dárky Leoně. Každý z nich má rád sebe sama a Karla.



obr. 2.7

Struktura interpretace : **W** = {Jana, Karel, Leona},

Denotáty konstant : ***D***(jana) = Jana, ***D***(karel) = Karel, ***D***(leona) = Leona.

Denotáty predikátů : ***D***(**posílá_dárky**) = {(Jana, Karel), (Jana, Leona), (Karel, Leona)},

D(**má_rád**) = {(Jana, Jana), (Jana, Karel), (Karel, Karel), (Leona, Leona), (Leona, Karel)}

Ve struktuře interpretace příkladu 2.2 jsou všechny bázové vektory pravdivé. Totéž pak platí o síti jako celku.

2.2.3 Splnitelnost a platnost univerzálních sítí

V interpretaci s danou strukturou mohou být proměnným sítě přiřazeny jejich valuací (ohodnocením) nějaké hodnoty z universa diskursu **W**. Z univerzálního tvrzení se touto valuací stává bázové tvrzení, jehož pravdivost lze v dané struktuře interpretace určit.

Definice 2.4

Nechť *A* je univerzální síť, **S** je struktura její interpretace a ***v*** nějaká valuace všech proměnných z *A* v **S**. Nechť *A*[***v***] je výsledná síť valuace. Potom ***v*** *splňuje A* v **S**, právě když *A*[***v***] je pravdivá.

Definice 2.5

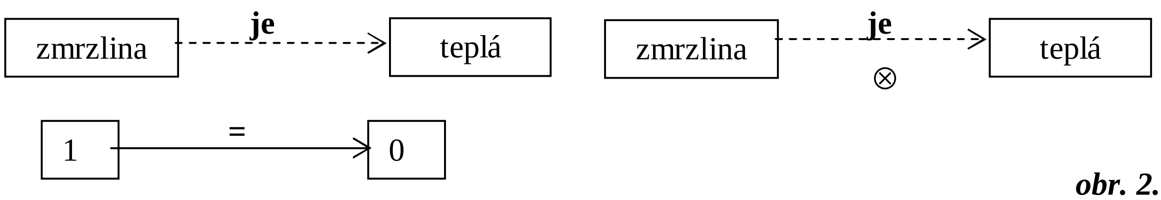
Univerzální asociativní síť (nepodmíněná síť nebo podmínka) *C* je *platná* ve struktuře **S**, právě když každá valuace proměnných z *C* splňuje *C* v **S**. Jinak je *C* nepravdivá v **S**. Struktura **S** potom síť *C splňuje*, je jejím *modelem*.

Struktura příkladu 2.2 je splňující strukturou, tj. modelem dané sítě, neboť síť, která nemá proměnné je též podle definice 2.5 platná v **S**.

2.2.4 Negace v asociativní síti

K popření bázového vektoru V se používá zvláštního obratu, tj. podmínky, která říká, že je-li V pravdivý, pak následuje nepravdivý konsekvent. Problémem je, jaké nepravdy lze použít pro účel popření - jde totiž o logickou nepravdu, tj. takovou, která je false ve všech interpretacích. Pro vyjádření, že neplatí tvrzení „Zmrzlina je teplá.“ není možno použít např. obratu „Je-li zmrzlina teplá, pak 1 = 0.“ (tj. reprezentace sítí vlevo), neboť při jiné interpretaci predikátu = nemusí konsekvent podmínku popírat.

Protože neexistují vektory typu kontradikce, je třeba pro účel popření zavést speciální prvek sítě zvaný falsum a označovaný ⊗, který je false v každé interpretaci (viz reprezentaci sítí vpravo).



obr. 2.8

Věta 2.1

Nechť V je bázový vektor. Pak v libovolné struktuře interpretace má podmínka s V jako antecedentem a ⊗ jako konsekventem opačnou pravdivostní hodnotu k V.

Důkaz

Předpokládejme, že podmínka V je true v nějaké struktuře **S**. Protože její konsekvent je v **S** false, musí její antecedent podle definice být též false v **S**. Předpokládejme naopak, že podmínka V je false v **S**. Pak protože její konsekvent je false v **S**, její antecedent V musí být true v **S** podle definice pravdivosti bázových podmínek.

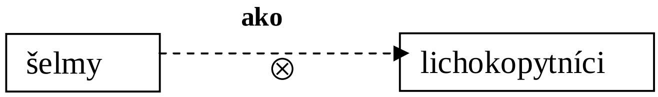
Definice 2.6

Podmínky, jejichž konsekventem je falsum, se nazývají *negativní tvrzení*.

Odvozovací metody se v asociativních sítích používají k dokazování tvrzení. Sémanticky to znamená, že tvrzení dokázané s využitím těchto dvou pravidel je pravdivé ve struktuře **S**, jsou-li tvrzení, z nichž se odvozovalo, pravdivá v **S**. Nyní bude uveden postup vyvracení tvrzení.

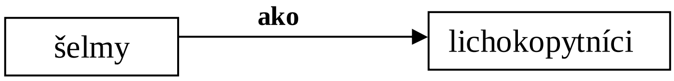
Příklad 2.3

V asociativní síti z příkladu 1.4 zde bude ukázáno, jak zavést do hlavní sítě negativní tvrzení, že šelmy nepatří mezi lichokopytníky, na základě dané podmínky reprezentované zvláštní sítí :



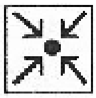
Postup spočívá v

1) Zařazení do asociativní sítě tvrzení, které má být v síti popřeno. V tomto případě jde o zařazení vektoru



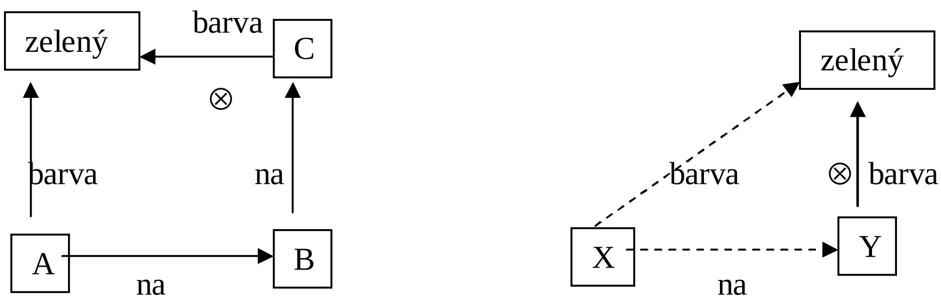
2) Použitím pravidla transferu k doplnění konsekventu ⊗ podmínky do hlavní sítě je odvozen spor s předpokladem zařazeným do sítě v kroku 1).

Příklad 2.4



Problém se týká světa tří bloků A,B,C. A leží na B, B na C. A je zelený, C není zelený a o barvě B není nic známo.

Je snadné ukázat, že v síti platí pravidlo : zelený blok leží na bloku, který není zelený.



obr. 2.9

2.2.5 Sémantická korektnost odvozování v asociativních sítích

Odvozování v asociativních sítích je postupem, jehož cílem je doplnit univerzální síť, která je zde sítí hlavní, při splnění nějakého pravidla reprezentovaného sítí zvláštní. Aby odvozování bylo korektním postupem, je třeba zaručit, že každá struktura splňující jak hlavní síť, tak i síť zvláštní, splňovala i síť odvozenou. Je proto třeba pro každé z odvozovacích pravidel dokázat jeho sémantickou korektnost.

Věta 2.2 (o sémantické korektnosti pravidla univerzální substituce)

Jestliže struktura S splňuje zvláštní síť *C* a existuje taková vazba vektorů z *C*, která vede ke shodě s vektory univerzální hlavní sítě, pak instance, která se odvodí z hlavní sítě uniformními substitucemi bázových termů sítě *C* za proměnné hlavní sítě, je rovněž strukturou S splněna.

Důkaz

Bázové termy na místě proměnných jsou prvky universa diskursu **W** struktury **S** a představují valuaci těchto proměnných univerzální sítě platné v **S**. Protože všechny proměnné hlavní sítě jsou univerzálního charakteru, je síť i touto valuací splněna.

Věta 2.3 (o sémantické korektnosti pravidla transferu)

Jestliže struktura S splňuje pravidlo C zvláštní sítě a každý z vektorů jeho antecedentu se objeví v hlavní síti jako pravdivý bázový vektor ve struktuře S, resp. stane se jím z univerzálního vektoru po aplikaci pravidla uniformní substituce, potom též konsekvent podmínky C, který byl pravidlem transferu přidán do sítě, je pravdivý ve struktuře S.

Důkaz vyplývá přímo z definice pravdivosti bázových podmínek.

Úkol 2.1

Do asociativní sítě reprezentující Vaše rodinné vztahy zařaďte tvrzení, že jste otcem/matkou syna/dcery. Na základě příslušné podmínky pak v síti odvoďte, že vaše matka/otec je babičkou/dědečkem

<u>Odvozováním v asociativních sítích</u> se rozumí doplňování dalších vektorů nebo termů v návěštích jejich uzlů do sítě, která je pro toto odvozování hlavní sítí, pomocí jiné zvláštní sítě některým z těchto odvozovacích pravidel :

<u>Pravidlo uniformní substituce</u> :

Jestliže se všechny ohodnocené hrany zvláštní sítě objeví v hlavní síti, obsahující proměnné jako návěští jimi spojených uzlů, pak lze návěští z uzlů zvláštní sítě substituovat uniformně za návěští odpovídajících uzlů sítě hlavní, označených proměnnými.

<u>Pravidlo transferu</u> :

Jestliže se všechny čárkované hrany antecedentu zvláštní sítě objeví (jako plné) v hlavní síti, přičemž návěští odpovídajících si uzlů se buď shodují (nebo jsou v některé ze sítí proměnnými), pak lze (event. po aplikaci pravidla uniformní substituce) plný vektor konsekventu ze zvláštní sítě doplnit do sítě hlavní.

<u>Strukturu interpretace</u> asociativní sítě tvoří podobně jako v predikátové logice trojice množin {**W**, **F**, **R**}, kde **W** je množina objektů světa, který má být asociativní sítí reprezentován, každé konstantě, existenční konstantě, n-ární funkci, n-árnímu existenčnímu funktoru a predikátu sítě je přiřazen denotát.

V případě konstanty je tímto denotátem objekt universa diskursu, v případě existenční konstanty je jím celé universum diskursu. V případě n-ární funkce je jím příslušný n-ární funkční předpis, v případě existenčního n-árního funktoru je jím množina všech n-árních funkcí nad W, a v případě predikátu je jím jeho interpretující relace.

Pravdivost bázových sítí

Bázový vektor **r** = AB je pravdivý ve struktuře **S** dané interpretace, jestliže denotát objektu A a denotát objektu B jsou v relaci R, která je denotátem predikátu **r** ve struktuře **S**. V opačném případě je nepravdivý.

Bázová nepodmíněná síť je pravdivá ve struktuře **S** dané interpretace, jsou-li v ní pravdivé všechny její bázové vektory.

Bázová podmínka je nepravdivá ve struktuře **S**, jestliže všechny její vektory antecedentu jsou pravdivé a vektor konsekventu je nepravdivý v **S**. Jinak je tato podmínka pravdivá.

Splňování a platnost univerzálních sítí

Nechť *A* je univerzální síť, **S** je struktura její interpretace a ***v*** nějaká valuace všech proměnných z *A* v **S**. Nechť *A*[***v***] je výsledná síť valuace. Potom ***v*** ***splňuje*** *A* v **S**, právě když *A*[***v***] je pravdivá.

Univerzální asociativní síť (nepodmíněná síť nebo podmínka) *C* je *platná* ve struktuře **S**, právě když každá valuace proměnných z *C* splňuje *C* v **S**. Jinak je *C* nepravdivá v **S**. Struktura **S** potom síť *C* ***splňuje***, je jejím ***modelem***.

Podmínky, jejichž konsekventem je falsum, se nazývají ***negativní tvrzení***.

Formální systém asociativních sítí je systémem sémanticky korektním a úplným.

Kontrolní otázky :

1. Kdy je bázový vektor pravdivý ?
2. Kdy je pravdivá bázová nepodmíněná síť ?
3. Kdy je pravdivá bázová podmínka ?
4. Kdy je splnitelná a kdy je platná univerzální nepodmíněná asociativní síť ?
5. Kdy je splnitelná a kdy je platná univerzální podmínka ?

Literatura

1. Brachman, R.J, Levesque, H.J. : Readings in Knowledge Representation. Morgan Kaufmann, Los Altos, 1985.
2. Ginsberg, Matt : Essentials of Artificial Intelligence Morgan Kaufmann Publishers, 1993
3. Kremer, R. : Visual Languages for Knowledge Representation. Proc. of KAW'98, Eleventh Workshop on Knowledge Aquisition, Voyager Inn, Banff, Alberta, Canada, 1998. http://www.hum.auc.dk/cg/
4. Lukasová, A. : Reprezentace Znalostí v Asociativních Sítích. Proc. Znalosti 2001,
5. Quillian, M.R. : Semantic memory. In Minsky, M. (ed.) : Semantic Information processing, MA : MIT Press, pp. 27-70, 1968.
6. Richards, T. : Clausal Form Logic. An Introduction to the Logic of Computes. Addison-Wesley, 1989.

3 NEMONOTÓNNÍ SYSTÉMY REPREZENTACE ZNALOSTÍ

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se dozvíte, že predikátová, resp. klauzulární logika se svou vysokou mírou expresivity je sice všeobecně uznávaným prostředkem reprezentace znalostí, ale ve srovnání s tím, jak lidský rozum manipuluje se znalostmi za účelem odvozování a vytváření teorií, trpí obě tyto logiky závažnými nedostatky, přinejmenším tím, že co jednou bylo v teorii odvozeno, nedá se revidovat, což mnohdy vede ke sporným teoriím.

3.1 MONOTÓNNÍ A NEMONOTÓNNÍ ODVOZOVÁNÍ

3.1.1 Monotónní odvozování

Znalostní systémy založené na tradiční logice prvního řádu pracují tak, že výchozí znalosti jsou zapsány ve formulích příslušného formálního jazyka tvořících znalostní bázi a systém pak rozhoduje s použitím svých odvozovacích pravidel, zdali je nebo není formule logickým důsledkem znalostní báze, resp. generuje formálními prostředky další logické důsledky této znalostní báze. Tento model je v mnoha případech užitečný, ale má některá závažná omezení, neboť existují souvislosti, pro něž monotónní odvozovací pravidla nevystihují myšlení, tak jak probíhá ve skutečnosti.

- Odvozovací pravidla nikdy neodvodí novou znalost o modelovaném světě. Odvození důsledku daných předpokladů pouze způsobí, že se znalostem skrytým implicitně v předpokladech dostane explicitního vyjádření.
- Jestliže je věta dokázána (viz dále příklad 3.1) pouze na základě části znalostní báze, nezjistí se její neplatnost (spor) při použití celé znalostní báze.
- Jazykem logiky prvního řádu nelze zachytit všechno, co je potřeba říci o modelovaném světě, zejména s ohledem na průběh dalších poznání o něm. Ne všechna odvozování, která je třeba v rámci umělé inteligence formalizovat, jsou trvale platná. Často se stává, že inteligentní systémy pracují s předpoklady, které jsou všeobecně přijímány, ale které mohou být později vyvráceny nebo modifikovány. Důsledky se dodáním dalších informací o stavu modelovaného světa mohou stát neplatnými a je třeba je revidovat.
- Logika prvního řádu totiž neumožňuje formulovat tvrzení alespoň s takovou mírou neurčitosti, která by dávala možnost revize již odvozených tvrzení na základě nových znalostí dodaných do znalostní báze.

Vlastnost formulovaná ve druhém bodě je důsledkem vlastnosti ***monotónnosti*** tradičních formálních systémů logiky prvního řádu. Dedukce v logice prvního řádu vychází z určité znalostní báze $\mathbf{S} = \{P_1, P_2, \ldots, P_n\}$, z níž je pak pomocí odvozovacích pravidel odvozena formule A. Stejně dobře však lze A odvodit z nadmnožiny $\mathbf{R}$. Pro všechny zde diskutované formální teorie totiž platí v logice prvního řádu známá *věta o rozšíření množiny předpokladů*, která tvrdí, že platí-li

$$P_1, P_2, \ldots, P_n \vdash A,$$

pak platí i

$$P, P_1, P_2, \ldots, P_n, \vdash A.$$

To je důsledkem ***kompaktnosti*** logiky prvního řádu, která tvrdí, že $\mathbf{S} \vdash A$, neboť pro nějakou konečnou podmnožinu $\mathbf{S}_A \subseteq \mathbf{S}$ platí $\mathbf{S}_A \vdash A$. Ostatní formule, které nevstupují do důkazu, nejsou předmětem zájmu (viz následující příklad). Tato vlastnost se nazývá vlastností ***lokality*** tradiční formální logiky.

Příklad 3.1

Nechť **S** = {**má_rád**(bb, zvířata), **žena**(bb), **herec**(bb, slavný), ∀x **má_rád**(x, zvířata) → ¬**nosí**(x, kožešiny), ∀x (**herec**(x, slavný) → **bohatý**(x)} je množina formulí (speciálních axiómů) tvořících znalostní bázi v některém z uvažovaných axiómatických systémů.

Formálně odvozenou větou teorie ze speciálních axiómů **S** (logickým důsledkem **S**) je zde mimo jiné formule

¬**nosí**(bb, kožešiny).

K formálnímu odvození této věty (důsledku) ze znalostní báze **S** ale stačí její podmnožina **S‘**⊂ **S**, kde

S‘ = {**má_rád**(bb, zvířata), ∀x **má_rád**(x, zvířata) → ¬**nosí**(x, kožešiny)}.

Charakteristickou vlastností tradičních axiómatických systémů je to, že jejich teorie (množiny logických důsledků speciálních axiómů) rostou *monotónně* s množinou formulí zařazených dodatečně do výchozí znalostní báze. Při monotónním odvozování přidání formule do znalostní báze nikdy neznehodnotí pravdivost jejích předcházejících důsledků. Tato vlastnost však je evidentně v rozporu s představou formální reprezentace vývoje poznání.

Následující věta shrnuje známé vlastnosti formálních systémů logiky prvního řádu.

Věta 3.1

Tradiční formální systémy logiky prvního řádu se vyznačují následujícími vlastnostmi ($P_1, P_2,..., P_n, A, Q, R$ jsou formule), které zajišťují jejich požadované schopnosti :

1. Reflexivnost :
   $\{P_1, P_2,..., P_n, A\} \vdash A$,
   která zajišťuje možnost odvodit důsledek totožný s některým z předpokladů.
2. Monotónnost :
   je-li $\{P_1, P_2,..., P_n\} \vdash A$, pak je i $\{P_1, P_2,..., P_n, R\} \vdash A$, která zajišťuje, že odvozený důsledek nebude znehodnocen další dedukcí ani při rozšíření znalostní báze.
3. Tranzitivnost :
   je-li $\{P_1, P_2,..., P_n\} \vdash R$ a $\{P_1, P_2,..., P_n, R\} \vdash Q$ pak je i $\{P_1, P_2,..., P_n\} \vdash Q$, která zajišťuje, že mezivýsledky lze využít pro další odvozování.

Všechny tradiční typy logických formálních systémů generují pouze teorie těchto uvedených vlastností.

Definice 3.1 (monotónnosti formálního systému)

Formální systém je *monotónní*, jestliže v něm přidání nové znalosti o modelovaném světě do znalostní báze zvětší nebo alespoň nezmenší množinu odvozených důsledků.

Následující jednoduchý příklad řešený ve výrokové logice ukazuje, jak lze korektními důkazovými kroky v monotónním systému odvodit spor.

Příklad 3.2

Rodinu tvoří otec Antonín, matka Berta a tři děti : Cyril, Dáša a Eliška. Situace, jež v rodině nastaly při sledování televize, lze vystihnout takto :

1) Dívá-li se Antonín, dívá se i jeho žena.

2) Buď Dáša nebo Eliška nebo obě se dívají na televizi.

3) Dívá se buď Berta nebo Cyril, ale nikdy oba společně.

4) Dáša a Cyril se vždy buď oba dívají nebo oba nedívají.

5) Dívá-li se Eliška, dívá se též Antonín a Dáša.

Kdo se za těchto podmínek dívá na televizi ?

1. $a \rightarrow b$
2. $d \vee e$
3. $(b \vee c) \ \& \ \neg(b \ \& \ c)$
4. $(d \leftrightarrow c)$
5. $e \rightarrow a \ \& \ d$

Formule vyjadřující daný problém je zde třeba pro účel důkazu v hilbertovském systému H převést do tvaru vhodného pro dokazování. Prvních 8 řádků důkazu pak tvoří množina předpokladů dedukce **U** = { $a \rightarrow b$, $\neg d \rightarrow e$, ...}

1. $\mathbf{U} \vdash a \rightarrow b$		
2. $\mathbf{U} \vdash \neg d \rightarrow e$		
3. $\mathbf{U} \vdash \neg b \rightarrow c$		
4. $\mathbf{U} \vdash b \rightarrow \neg c$		
5. $\mathbf{U} \vdash c \rightarrow d$		
6. $\mathbf{U} \vdash d \rightarrow c$		
7. $\mathbf{U} \vdash e \rightarrow a$		
8. $\mathbf{U} \vdash e \rightarrow d$		
9. $\mathbf{U}, b \vdash \neg c$	dedukce na 4	Cyril se nedívá
10. $\vdash (d \rightarrow c) \rightarrow (\neg c \rightarrow \neg d)$	axióm (A3)	
11. $\mathbf{U} \vdash \neg c \rightarrow \neg d$	MP na 6, 10	
12. $\mathbf{U}, b \vdash \neg d$	MP na 9, 11	Dáša se nedívá
13. $\mathbf{U}, b \vdash e$	MP na 2, 12	Eliška se dívá
15. $\mathbf{U}, b \vdash d$	MP na 8, 13	Dáša se dívá

V 15. řádku důkazu, v jehož průběhu bylo použito korektní pravidlo dedukce, byl odvozen spor s řádkem 12. Stalo se tak zjevně proto, že použitím pravidla dedukce byla množina předpokladů rozšířena tak, že se stala nekonsistentní. Korektně odvozenými důsledky takovéto množiny předpokladů pak mohou být libovolné formule, tedy d i ¬d.

Aby se přiblížila skutečnému myšlení, musí formální logika vzít v úvahu i myšlení s neúplnými nebo nespolehlivými informacemi a především *revidovatelné myšlení*. Pak je ale třeba použít i *nemonotónních* prostředků odvozování.

3.1.2 Nemonotónní odvozování

Nemonotónní odvozování je takové odvozování, v němž množina závěrů neroste monotónně se vzrůstající množinou předpokladů.

Aby byla zajištěna nemonotónnost systému, je třeba podmínky definovat poněkud měkčeji.

Cílem nemonotónního odvozování je docílit co nejvyšší shody se skutečným všeobecně sdíleným usuzováním (commonsense reasoning), zahrnující schopnost docílit smysluplných závěrů i s neúplnými nebo nejistými znalostmi. Pochopení a zavedení principů skutečného všeobecně sdíleného usuzování do znalostních bází by mělo zahrnovat též implementaci odvozování založeného na nemonotónním přístupu. Na nemonotónní logiku přitom lze nahlížet jako na způsob rozšíření tradiční logiky o daný *kontext*.

Příklad 3.3

Příklad, kdy lze po doplnění znalostní báze o další znalosti v predikátové logice odvodit spor.

1. „Ptáci létají."
2. „Tweety je pták."

Závěr : „Tweety létá."

Přidáním dalších předpokladů :

3. „Tučňáci jsou ptáci."
4. „Tučňáci nelétají."
5. „Tweety je tučňák."

Závěr 2 : „Tweety nelétá."

Z původních dvou formulí lze vytvořit v predikátové logice :znalostní bázi

1. $\forall x\ (\textbf{pták}(x) \rightarrow \textbf{létá}(x))$
2. $\textbf{pták}(\text{Tweety})$

   Z ní lze odvodit

   $\textbf{létá}(\text{Tweety})$

   Po doplnění znalostní báze o další tři formule
3. $\forall x\ (\textbf{tučňák}(x) \rightarrow \textbf{pták}(x))$
4. $\forall x\ (\textbf{tučňák}(x) \rightarrow \neg\textbf{létá}(x))$
5. $\textbf{tučňák}(\text{Tweety})$

   lze odvodit

   $\neg\textbf{létá}(\text{Tweety})$

Premisa tvrdící, že Tweety je tučňák se zdá být relevantnější než ta, která tvrdí, že Tweety je pták. Je to tím, že všichni tučňáci jsou ptáci (premisa 3.), nikoliv naopak. Premisa o tučňácích je více specifická než premisa o ptácích. Tady lze spatřovat měřítko relevantnosti argumentů. Máme-li snahu využívat především ty nejrelevantnější, odvodíme, že Tweety nelétá. Premisa 3, která vyjadřuje všeobecně sdílenou znalost, je zde klíčová. Premisa 1 je zde pravidlem s výjimkami, které zatím neznáme. Tak ale probíhá skutečné myšlení – činíme závěry, aniž bychom čekali na příchod dalších informací. Jinými slovy, přijímáme předpoklad uzavřeného světa.

Srovnáme-li obě premisy o létání, jedna se týká ptáků, druhá jejich podmnožiny – tučňáků. Tato diference ve specifičnosti určuje i relevantnost argumentů. Specifičtější tvrzení je více relevantní, proto „Tweety nelétá." Autoři práce [3] na hierarchii specifičností argumentů zakládají způsob rozhodování, zdali odvozený důsledek bude zahrnut do rozšíření teorie. Další příklad ukazuje, že někdy nelze takto uvažovat, neboť specifičnosti nelze vždy uspořádat.

Příklad 3.4

Jde o z literatury známý příklad, v němž nelze na základě informace (1 – 4) o příslušnosti k politické orientaci a jejích charakteristických rysech odvodit, zdali bývalý prezident Richard Nickson má povahu holubičí nebo jestřábí.

1. „Kvakeři jsou holubi."
2. „Republikáni jsou jestřábi."
3. „Dick je republikán."
4. „Dick je kvaker."

1. $\forall$x (**kvaker**(x) $\rightarrow$ **holub**(x))
2. $\forall$x (**republikán**(x) $\rightarrow$ **jestřáb**(x))
3. **republikán**(Dick)
4. **kvaker**(Dick)
   Z těchto předpokladů lze odvodit
5. **holub**(Dick)
6. **jestřáb**(Dick)

Chybí informace o hierarchii specifičnosti premis, proto nelze přijmout ani „Dick je holub." ani „Dick je jestřáb."

Definice 3.2 (nemonotónnosti formálního systému)

Formálního systém je *nemonotónní*, jestliže připouští, aby přidání nové znalosti o modelovaném světě do znalostní báze znehodnotilo některé jeho již odvozené věty.

Základními typy nemonotónních logických systémů, které představují jisté přiblížení se všeobecně sdílenému uvažování, jsou podle řady autorů :

A. Reiterova default logika
B. Nemonotónní logika podle McDermotta
C. Autoepistemická logika
D. Techniky založené na Closed world assumption a circumscription

Poznámka :

Jedním z cílů teorie nemonotónního odvozování je řešení tzv. všeobecně sdíleného zákona setrvačnosti. Tento zákon formuloval Leibnitz takto : „ Všechno má tendenci setrvat v tom stavu, ve kterém se právě nachází." Přesný formální zápis tohoto zákona je cílem mnoha snažení.

3.2 FORMALIZACE REVIDOVATELNÉHO ODVOZOVÁNÍ

Většina přístupů k nemonotónnímu odvozování vychází z tradiční logiky 1. řádu. Existují v podstatě dvě cesty, jak rozšířit tradiční predikátovou logiku tak, aby zahrnovala neurčitost umožňující nemonotónní odvozování :

- rozšíření jazyka
- zavedení dalších odvozovacích pravidel

3.2.1 Řešení pomocí rozšíření jazyka predikátové logiky

Přístup označovaný v literatuře jako circumscription (vymezení) formalizuje pojmy, jakými jsou např.normálnost a typičnost pomocí

- standardního predikátu **normální** :

$$\forall x\ (\mathbf{pták}(x)\ \&\ \mathbf{normální}(x) \rightarrow \mathbf{létá}(x))$$

- nového logického operátoru typicky

$$\forall x\ (\mathrm{typicky}(\mathbf{pták}(x)) \rightarrow \mathbf{létá}(x)).$$

Další možností je též využití specifikujících predikátů k vyjmenování případů, kdy tvrzení neplatí :

$$\forall x\ (\mathbf{pták}(x)\ \&\ \neg(\mathbf{tučňák}(x) \vee \mathbf{poraněný}(x) \vee ...) \rightarrow \mathbf{létá}(x))$$

Nevýhodou poslední z vyjmenovaných možností je nutnost přepracovat formuli po každém zjištění další výjimky.

Směry, které používají např. predikát **normální** v syntaxi

$$\forall x\ \mathbf{normální}(\mathbf{pták}(x) \rightarrow \mathbf{létá}(x)),$$

posouvají formální reprezentaci do oblasti logiky druhého řádu.

Další v literatuře uváděnou možností je doplnění logiky prvního řádu o standardní predikát **unless**. Predikát **unless**(*P*) je true pro formuli *P*, právě tehdy, nelze-li *P* odvodit z dané množiny předpokladů **A**. Tento predikát lze, jak bude blíže vysvětleno dále, použít v definicích odvozovacích pravidel, která umožňují odvodit nové tvrzení za podmínky, že jiné tvrzení odvodit nelze. Deduktivní systém výrokové logiky doplněný pravidly obsahujícími predikát **unless** je sice nemonotónní, ale nemá požadované vlastnosti. Může se totiž stát, jak bude ukázáno dále, že *P* a **unless**(*P*) bude odvozeno zároveň.

3.2.2 Reiterovo řešení pomocí dalších odvozovacích pravidel

Jedním ze způsobů realizace nemonotónního odvozování je definování logiky, která vedle formulí tradiční logiky používá nestandardní nemonotónní odvozovací pravidla nazývaná *default pravidly*. Přístup *default logiky*, která přidává k logice prvního řádu default odvozovací pravidla, představuje druhou možnost rozšíření logiky prvního řádu.

Default logika byla uvedena Reiterem jako technika umožňující vyjádřit a odvozovat na základě zákonitostí, které platí až na některé výjimky. Tento způsob dedukce se nazývá *default odvozování*. Řada dalších nemonotónních logik se vymezuje právě vůči Reiterově *default* logice.

Definice 3.3 (default pravidla)

Nechť $\alpha = \alpha_1, ..., \alpha_m$ $\beta = \beta_1, ..., \beta_n$ a γ jsou dobře utvořené uzavřené formule logiky prvního řádu. Potom pravidlo

$$\frac{\alpha : \beta}{\gamma}$$

nebo též

$$(\alpha : \beta) \rightarrow \gamma,$$

„Platí-li α a je-li β konzistentní s tím, co je všeobecně považováno za platné, potom je možno usuzovat, že platí γ.“

je ***default pravidlo*** (stručně default).

$\alpha = \alpha_1, ..., \alpha_m$ představuje *prerekvizity* pravidla, $\beta_1, ..., \beta_n$ jsou jeho *oprávnění*, γ je *závěr*.

Speciální případy default pravidla :

- $(\alpha :) \rightarrow \gamma$
  znamená, že „když platí α, pak je možno usuzovat, že platí i γ “, což není totéž jako $\alpha \rightarrow \gamma$ (platí-li α, pak nutně platí i γ).
- $(: \beta) \rightarrow \gamma$
  znamená, že když je β konzistentní s tím, čemu se všeobecně věří, pak můžeme usuzovat, že platí γ.
- *normální* default pravidlo $(\alpha : \gamma) \rightarrow \gamma$
  znamená, že „platí-li předpoklady a závěr je konzistentní s tím, čemu se všeobecně věří, pak lze odvodit, že závěr platí.“ Např. (pták(x) : létá(x)) → létá(x) znamená, že „Je-li x pták a je-li tvrzení ‘x umí létat‘ konzistentní s tím, co je všeobecně známo, pak x umí létat.“

V předcházejícím odstavci uvedený predikát **unless** by se jevil vhodným pro formulaci oprávnění, kdyby v některých případech (viz následující příklad) nevedl ke sporným závěrům.

Příklad 3.5

Nechť **A** = {a, a & **unless**(b) → c, a & **unless**(c) → b}, kde a,b,c, jsou výrokové proměnné, je množina předpokladů. Následující dva tablové důkazy ukazují, že z **A** lze odvodit buď b nebo c, nikoliv však zároveň.

Tablový důkaz c vycházející z předpokladu, že nelze dokázat b, tj. pro případ, kdy **unless**(b) je true :

a, a & **unless**(b) → c, ¬c

a, a & T → c, ¬c

a, ¬(a & T), ¬c   a, c, ¬c

obr. 3.1

Protože c dokázat lze, je hodnota predikátu **unless**(c) false. Potom ale, jak ukazuje následující tablo, nelze dokázat b.

a, a & **unless**(c) → b, ¬b

a, a & F → b, ¬b

a, ¬(a & F), ¬b   a, b, ¬b

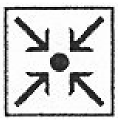
○   ×

obr. 3.2

3.2.3 Význam konsistence v nemonotónních systémech

Následující příklad důkazu logického důsledku ve výrokové logice na základě její sémantiky ve srovnání s důkazem podle default pravidla ukazuje zásadní význam konsistence v nemonotónním dokazování.

Příklad 3.6



"Podezřelý může být vinen pouze tehdy, byl-li letošního 1. ledna v 9 hodin ráno v Praze.", "Bylo zjištěno, že podezřelý byl v 9 hodin ráno v Ostravě." |= "Podezřelý je nevinen."

Zavedené označení :

"Podezřelý může být vinen." - v

"Podezřelý byl letošního 1. ledna v 9 hodin ráno v Praze." - p

"Podezřelý byl letošního 1. ledna v 9 hodin ráno v Ostravě." - o

Potom je

"Podezřelý může být vinen pouze tehdy, byl-li letošního 1. ledna v 9 hodin ráno v Praze." - $v \leftrightarrow p$

V daném případě jde o dedukci

$$v \leftrightarrow p, o \models \neg v$$

Snadno lze ověřit, že ¬v není logickým důsledkem množiny uvedených předpokladů. Ta má totiž dva modely a ¬v je splněna pouze v jednom z nich.

Při aplikaci normálního default pravidla jde o to dokázat :

$$\frac{v \leftrightarrow p, o \ : \ \neg v}{\neg v} \quad ,$$

což platí. ¬v proto patří do normálního default rozšíření teorie.

Z definice 3.6 i uvedeného příkladu je zde zřejmý zásadní význam pojmu konsistence. Na příkladě z výrokové bylo ilustrováno, co se v tradičních logických systémech může stát, je-li se konsistence množiny předpokladů ignorována. Podle platných odvozovacích pravidel lze v těchto tradičních systémech odvodit i nesmyslné sporné závěry.

Deduktivní systém formalizující nemonotónní uvažování musí být schopen odvodit přijatelné logické důsledky (označujeme |~). Ty ale musí být s množinou předpokladů konsistentní, tj. platné alespoň v jednom z jejích modelů.

Formulace konsistence v nemonotónních systémech mívá zpravidla vágní podobu, jakou je např.

„Je-li x pták a je-li tvrzení ‘x umí létat‘ konsistentní s tím, co je známo, pak x umí létat.“

Pro formalizaci odvozování na bázi konsistence je však potřebné její přesnější vyjádření. Jednou z možností je použití operátoru M – *je konsistentní s*....V takto rozšířeném jazyce je např. následující formule :

$$\forall x\ (\text{pták}(x)\ \&\ M\ \text{létá}(x) \rightarrow \text{létá}(x))$$

Oba uvedené formalismy se zdají být podobné, ale ve skutečnosti je expresivnější AEL.

Autoepistemická logika (AEL) rozšiřuje logiku prvního řádu o operátor L, který se čte jako *věří se, že....*
Přepis předcházejícího tvrzení do formule tohoto jazyka :

$$\forall x\ (\text{pták}(x)\ \&\ \neg L \neg \text{létá}(x) \rightarrow \text{létá}(x))$$

„Je-li x pták a jestliže nemám důvod věřit, že nelétá, potom odvozuji, že létá.“

Pojem konsistence množiny formulí bývá v logice prvního řádu zpravidla definován z hlediska sémantického. Z hlediska formálního pak je β je konsistentní s **S**, jestliže nelze dokázat **S** $\vdash \neg\beta$. Předcházející vágní vyjádření konsistence pak lze zpřesnit :

„Je-li x pták a jestliže nelze dokázat, že x neumí létat, pak x umí létat.“

Příklad 3.7

„Je-li x pohárovitá jarní květina a je-li konsistentní podle všeho,co je v současnosti známo, předpokládat, že x je krokus, pak odvoď, x je krokus.“

Uvedená věta se přepíše v této notaci takto :

$$\frac{\text{pohárovitá\_jarní\_květina}(x) :\ \text{krokus}(x)}{\text{krokus}(x)}$$

Předpokládejme dále, že máme pravidlo, které stanoví, že má-li pohárovitá jarní květina šest tyčinek, je to určitě ocún.

$$\forall x\ (\text{pohárovitá\_jarní\_květina}(x)\ \&\ \text{má\_šest\_tyčinek}(x) \rightarrow \text{ocún}(x)).$$

Jestliže odvodíme podle předcházejícího pravidla, že x je krokus a později si všimneme šesti tyčinek, je třeba revidovat přecházející závěr. Nadále totiž není konsistentní předpokládat, že jde o krokus a default pravidlo tedy již není aplikovatelné. Důležité je proto definovat metodu budování *rozšíření teorie* způsobem, který umožňuje toto nemonotónní chování.

Tento příklad demonstruje model skutečného myšlení. Nějaký okamžitý dojem (prerekvizita) vede bezprostředně k závěru, který může být založen na předcházející zkušenosti. Tento závěr však byl učiněn na neúplné informaci. Tak se to děje i ve skutečnosti. Lze totiž jednat i bez předcházejícího shromáždění všech relevantních informací.
Následující příklady ukazují různé formy nemonotónního odvozování vyjádřeného pomocí defaultů.

Příklad 3.8

V každém z těchto příkladů default pravidlo [α:β/γ] čteme jako „Je-li α true, přičemž je možné věřit β, pak závěr γ je true. “

a) Typické děti mají dva rodiče. To lze vyjádřit pomocí defaultů takto :

$$\frac{\text{dítě}(x)\text{: má\_dva\_rodiče}(x)}{\text{má\_dva\_rodiče}(x)}$$

To lze číst takto : „ Pokud je rozumné věřit, že dítě má dva rodiče, pak tomu věřme.“

b) - odvozování bez rizika

Předpokládáme, že obžalovaný je nevinný, pokud se nedokáže opak.

$$\frac{\text{obžalovaný}(x) : \text{nevinný}(x)}{\text{nevinný}(x)}$$

c) – nejlepší odpověď

Předpokládáme, že nejlepší nalezené řešení je tím nejlepším řešením.

$$\frac{\text{předpokládané\_nejlepší\_řešení}(x) : \text{nejlepší\_řešení}(x)}{\text{nejlepší\_řešení}(x)}$$

d) – pravidlo Sherlocka Holmese

$$\frac{\text{jediné\_zbývající řešení}(x) : \text{možné\_řešení}(x)}{\text{řešení}(x)}$$

e) – komunikační konvence

Jestliže vlak z Londýna do Glasgow v 10.30 není v jízdním řádu, pak je bezpečné předpokládat, že žádný vlak z Londýna do Glasgow v 10.30 nejede.

$$\frac{\neg\text{je\_v\_jízdním\_řádu}(x) : \neg\text{odjíždí\_v}(x)}{\neg\text{odjíždí\_v}(x)}$$

Tento typ negace je znám z Prologu, který nepoužívá klasickou negaci, ale negaci selháním důkazu. tj., jestliže interpreter Prologu není schopen dokázat, že klauzule je důsledkem programu, pak ji považujeme za nepravdivou.

3.2.4 Pojem omezené monotónnosti

V literatuře jsou uváděny dva přístupy k tzv. omezené monotónnosti :

1. *Podmiňující monotónnost* :

Jestliže $\{P_1, P_2,..., P_n\}|\!\sim R$ a zároveň $\{P_1, P_2,..., P_n\}|\!\sim Q$, pak $\{P_1, P_2,..., P_n, R\}|\!\sim Q$.

2. *Racionální monotónnost* :

Jestliže neplatí $\{P_1, P_2,..., P_n\}|\!\sim \neg R$ a zároveň platí $\{P_1, P_2,..., P_n\}|\!\sim Q$, pak $\{P_1, P_2,..., P_n, R\}|\!\sim Q$.

Ztrácí se jednoduchá iterativní struktura klasických axiómatických systémů, která umožňuje konstruovat a spočítat množinu možných odvozených logických důsledků. Je však možno charakterizovat množinu odvozených formulí pomocí pevných bodů. „Pevným bodem“ je stabilní množina formulí, vzhledem k níž již nemůže být žádná další formule odvozena konsistentně.

V klasické logice je odvozená věta tautologií. V souvislosti s nemonotónními logikami je třeba nejdříve definovat jako věty ty formule, které se objevují ve všech odvoditelných stabilních množinách. Tyto formule tvoří jedinou množinu. V tomto pojetí důkaz věty spočívá v prověření, že se tata formule vyskytuje ve všech stabilních množinách – pevných bodech.

Náhradou požadavku platnosti požadavkem konsistentnosti v nemonotónních logikách ztrácí platnost řada vět klasické logiky, např. věta o dedukci (srovnej s uvedeným příkladem 3.2).

Σ Formální systém je *monotónní*, jestliže v něm přidání nové znalosti o modelovaném světě do znalostní báze zvětší nebo alespoň nezmenší množinu odvozených důsledků. Formálního systém je *nemonotónní*, jestliže připouští, aby přidání nové znalosti o modelovaném světě do znalostní báze znehodnotilo některé jeho již odvozené věty.

Většina přístupů k nemonotónnímu odvozování vychází z tradiční logiky 1. řádu. rozšířené tak, aby zahrnovala neurčitost umožňující nemonotónní odvozování :

- rozšířením jazyka
- zavedením dalších odvozovacích pravidel

Rozšíření jazyka spočívá např. v zavedení standardního predikátu **normální**

$$\forall x\ (\mathbf{pták}(x)\ \&\ \mathbf{normální}(x) \rightarrow \mathbf{létá}(x))$$

nebo nového logického operátoru typicky

$$\forall x\ (\text{typicky}(\mathbf{pták}(x)) \rightarrow \mathbf{létá}(x)).$$

Další možností je též využití specifikujících predikátů k vyjmenování případů, kdy tvrzení neplatí :

Druhou možnost rozšíření logiky prvního řádu představuje přístup *default logiky*, která přidává k logice prvního řádu default odvozovací pravidla :

Nechť $\alpha = \alpha_1, ..., \alpha_m$ $\beta = \beta_1, ...,\beta_n$ a γ jsou dobře utvořené uzavřené formule logiky prvního řádu. Potom pravidlo

$$\frac{\alpha : \beta}{\gamma}$$

nebo též

$$(\alpha : \beta) \rightarrow \gamma,$$

„Platí-li α a je-li β konsistentní s tím, co je všeobecně považováno za platné, potom je možno usuzovat, že platí γ.“

je *default pravidlo* (stručně default).

$\alpha = \alpha_1, ..., \alpha_m$ představuje *prerekvizity* pravidla, $\beta_1, ...,\beta_n$ jsou jeho *oprávnění*, γ je *závěr*.

Speciálním případem default pravidla je *normální* default pravidlo $(\alpha : \gamma) \rightarrow \gamma$ znamená, že „platí-li předpoklady a závěr je konsistentní s tím, čemu se všeobecně věří, pak lze odvodit, že závěr platí.“

Deduktivní systém formalizující nemonotónní uvažování musí být schopen odvodit přijatelné logické důsledky (označujeme |~). Ty ale musí být s množinou předpokladů konsistentní, tj. platné alespoň v jednom z jejích modelů.

Kontrolní otázky :

1. V čem se liší monotónní a nemonotónní formální systémy ?
2. Proč nemonotónní odvozování lépe odpovídá skutečnému usuzování ?
3. Jakým způsobem lze modifikovat logiku prvního řádu, aby umožňovala nemonotónní odvozování ?
4. Co jsou default pravidla ?
5. Jaký význam má konsistence v nemonotónním odvozování ?

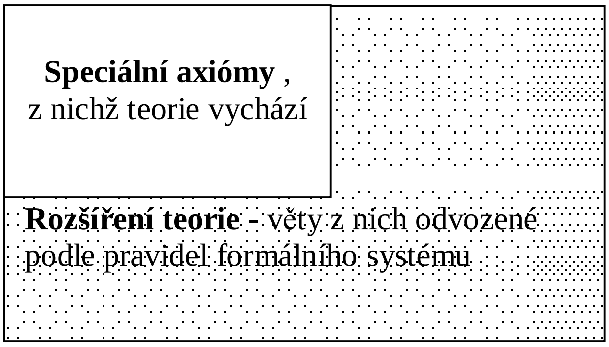
4 BUDOVÁNÍ REVIDOVATELNÝCH TEORIÍ

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se seznámíte s default systémy nemonotónního odvozování a budování teorií, které jsou všeobecně považovány za nejperspektivnější. Seznámíte se též s formální definicí default_negace teorie

4.1 NEMONOTÓNNOST V BUDOVÁNÍ TEORIÍ

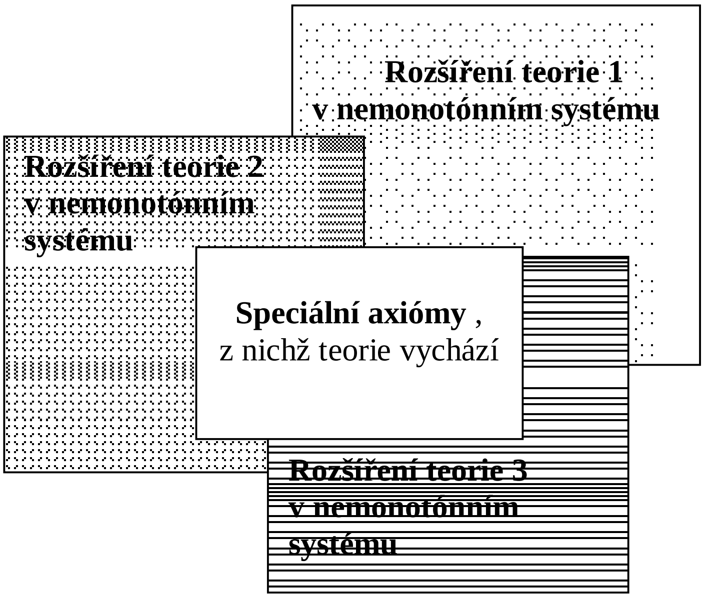
4.1.1 Budování teorií v monotónních a nemonotónních systémech

Teorie je v monotónních systémech budována na základě konsistentní množiny speciálních axiómů tak, že do jejího rozšíření náleží každé odvozené tvrzení (věta) ze speciálních axiómů a též další věty odvozené z již odvozených vět. Přidáváním dalších později získaných tvrzení, která se ukázala jako pravdivá, neovlivní nijak obsah až dosud vybudované teorie (obr.4.1). Jestliže nově dodaná pravdivá tvrzení způsobí, že se rozšířená teorie stane spornou, je možno v případě automatizovaným generováním dalších vět bez kontroly konsistence vygenerovat spousty nesmyslů.



obr. 4.1

Nemonotónní systémy berou v úvahu skutečnost, že do základní množiny speciálních axiómů teorie mohou náležet i tvrzení, která jsou důsledky empirických generalizací. Tato provizorní tvrzení a v důsledku i věty z nich odvozené mohou mít rizikovou existenci. Jednou vybudované rozšíření teorie tak nemusí mít věčnou platnost. Dodání dalších opravných tvrzení, získaných o zmíněných předmětech empirického zkoumání, může způsobit znehodnocení celého nebo části původního rozšíření teorie a vybudování nového rozšíření na základě opravené množiny poznatků (obr. 4.2).



obr. 4.2

V reprezentaci znalostí se formální logika prakticky omezuje na relaci důsledku, ať už se jedná o její monotónní nebo nemonotónní variantu.

Je třeba zavést speciální formální prostředky, tj. jazyk a odvozovací pravidla, pro případy, kdy jsou odvozené důsledky bezpečné, ale též pro případy, kdy tomu tak není.

Jsou to dva druhy tvrzení, které způsobují nemonotónnost :

- empirická generalizace (speciální axiómy s možnými výjimkami)
- provizorně odvozené věty (věty odvozené z těchto empirických generalizací)

Generalizaci je třeba pojímat tak, aby byla informačně stabilní v tom smyslu, že odvozená rozšíření ji neovlivní. Jediné, co může generalizaci postihnout, je omezení její aplikovatelnosti, tedy užší specifikace, doprovázená event. novými specifickými pravidly.

4.2 FORMALIZACE BUDOVÁNÍ DEFAULT TEORIE

V následujících definicích podle Reitera se předpokládá, že pravidla a všechny formule jsou vytvořeny z uzavřených formulí logiky prvního řádu.

Definice 4.1 (default teorie)

Default teorie $\Delta = (\mathbf{T}, \mathbf{D})$ je dvojice množin, kde **T** je množinou uzavřených formulí logiky prvního řádu a **D** je množina default pravidel.

Definice 4.2 (normální default teorie)

Default teorie $\Delta = (\mathbf{T}, \mathbf{D})$ je *normální*, jsou-li všechna její default pravidla z **D** normální.

Definice 4.3 (default rozšíření)

Množina důsledků default teorie se nazývá jejím *rozšířením*.

4.3 VLASTNOSTI NEMONOTÓNNÍCH TEORIÍ

4.3.1 Revidovatelnost

V nemonotónních formálních systémech musí být možné v některých případech stáhnout zpět již odvozený důsledek. K tomu je potřeba revizních odvozovacích pravidel, tj. pravidel dynamického charakteru, která umožní ještě před odvozením verifikovat, že tvrzení, které má být odvozeno, je konsistentní se vším, co již předtím bylo v systému z premis odvozeno. Tato verifikace se může dynamicky měnit s narůstající množinou formulí již v systému odvozených.

Všechny systémy založené na empirické generalizaci nebo víře jsou náchylné k nekonsistenci. Každý formální způsob nemonotónního odvozování musí obsahovat postup, jak naložit s nekonsistencí – jak ji rozpoznat, jak ji vyřešit a jak rozlišit správné odvození v rámci nekonsistentního systému od nesprávného.

4.3.2 Regulárnost, založenost a saturovanost rozšíření (extenze)

Rozšíření teorie, které je vybudováno tak, aby zůstalo konsistentní i po přidání všech odvozených závěrů, se nazývá *regulární*. Požadavek regulárnosti dělá je třeba ošetřit zejména při automatizovaném dokazování. Při aplikaci každého pravidla se zajištěním regularity je třeba kontrolovat konsistenci s celým rozšířením znalostní báze.

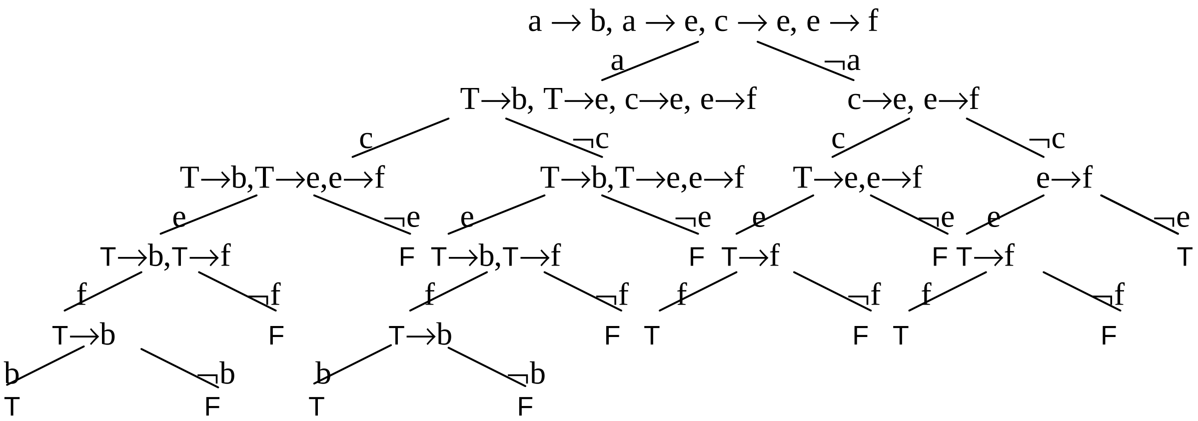
Požadujeme-li, aby každá formule z rozšíření byla odvoditelná na základě skutečností tvrzených ve výchozí množině speciálních axiómů, jde nám o jeho založenost (groundedness).

Před formální specifikací rozšíření je třeba si uvědomit ještě požadavek, aby všechna aplikovatelná pravidla byla aplikována, tj. aby rozšíření bylo *saturováno*.

Příklad 4.1

Nechť je dána default teorie Δ = (**T**, **D**) = ({a → b, a → e, c → e, e → f},
{: ◊(a)/a, : ◊(c)/c}), pro niž je
T = {a → b, a → e, c → e, e → f} **(T** je konsistentní množina, jak lze snadno dokázat), **D** = {(:a)/a, : (:c)/c}), tj. teorie se dvěma defaulty, které říkají, že když je konsistentní, že platí a, pak a považujeme za pravdivé, podobně pro c.

Je zřejmé, že s danou konsistentní množinou **T** je a i c konsistentní, proto jsou odvoditelné daným default pravidelem. V prvním případě to dokazuje např. model {a, b, c, e, f, b}, ve druhém např. model {¬a, b, c, e, f}. Množina literálů {b, e, f} je zde společná všem modelům a tvoří tak „pevný bod“.



obr. 4.3

Na tomto příkladě je vidět, že jsou zde dva přístupy k sémantice default teorie.

- Výběr důsledků default pravidel na základě konsistence (modelů).
- Jinou volbou je snažit se najít množiny výroků, které platí bez ohledu na to, která množina konsistentních default pravidel bude použita. V našem příkladě je to množina {b, e, f}.

4.4 DEFAULT-NEGACE

Konsistenci formule *A* s množinou formulí **S** lze v logice prvního řádu zajišťovat na základě

- sémantického hlediska, tj. prověřovat existenci modelu množiny formulí $A \cup \mathbf{S}$,
- nebo hlediska formálního, neboť je *A* konsistentní s **S**, jestliže nelze dokázat $\mathbf{S} \vdash \neg A$.

Právě druhý z uvedených způsobů sledování a zajišťování konsistence při rozšiřování teorií je základem tzv. Reiterova default-negace přístupu.

Definice 4.4

Default-negace teorie je dána dvojicí sestávající z množiny tvrzení **T** a množiny pravidel **D**. Každé pravidlo v **D** je specifikováno jako trojice (<pozitivní antecedenty>, <negativní antecedenty>, <konsekvent>).

Default-negace pravidlo má obecně tvar

$$\frac{\alpha_1 \,\&....\&\, \alpha_m \;:\; \sim\beta_1\&....\&\sim\beta_n}{\gamma}$$

Podmínky pravidla α_1 &....& α_m , $\sim\beta_1$&....& $\sim\beta_n$ jsou splněny vzhledem k množině předpokladů E (současnému stavu rozšíření) a lze odvodit závěr γ, jestliže všechny pozitivní antecedenty jsou přítomny v E a žádný negativní antecedent není přítomen v E.

Příklad 4.2

Vyjádření pravidla pro lékařskou praxi

„Trpí-li pacient bolestí a je-li skutečností, že aspirin uvolňuje bolest a jestliže pacient není alergický na aspirin, pak odvoď závěr, že je možno předepsat pacientovi aspirin.“

Default-negace pravidlo :

$$\frac{\text{pacient\_trpí\_bolestí, aspirin\_uvolňuje\_bolest} \;:\; \neg\text{pacient\_alergický\_na\_aspirin}}{\text{předepsat\_pacientovi\_aspirin}}$$

V jazyce tradiční výrokové logiky je možno toto pravidlo vyjádřit pomocí implikace : pacient_trpí_bolestí & aspirin_uvolňuje_bolest & ¬pacient_alergický_na_aspirin → předepsat_pacientovi_aspirin

4.4.1 Default-negace extenze podle Reitera

Reiter předložil následující definici rozšíření default-negace teorie. $\boldsymbol{Th}_{L}(\mathbf{E})$ v ní označuje množinu všech důsledků dobře utvořených formulí logiky **L**. V každém kroku vytváření posloupnosti $\mathbf{E}_i$ se k teorii přiřazují všechny důsledky v klasickém smyslu a též všechny důsledky aplikovatelných default pravidel. Definice proto neobsahuje saturační podmínku.

Definice 4.5

Nechť $\Delta = (\mathbf{T}, \mathbf{D})$ je uzavřená default teorie. Potom množina uzavřených dobře utvořených formulí **E** je Reiterovou extenzí teorie Δ, právě když

1) $\mathbf{E} = \bigcup_{i=0}^{\infty} \mathbf{E}_i$

kde

2) $\mathbf{E}_0 = \mathbf{T}$ a

$\forall i \;\; i \geq 1$

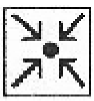
$\mathbf{E}_{i+1} = \boldsymbol{Th}_{L}(\mathbf{E}_i) \cup \{\gamma \mid [\alpha{:}\beta_1, ..., \beta_n/\gamma] \in \mathbf{D} \;\&\; \alpha \subseteq \mathbf{E}_i \;\& \neg\beta_1, ..., \neg\beta_n \notin \mathbf{E}\}$

V této definici je extenze definována tak, že je saturována a založena v **T**. Extenze je též regulární. To znamená, že do extenze jsou zařazeny pouze ty důsledky, jsou-li jejich oprávnění konsistentní s výslednou extenzí ($\neg\beta_1, ..., \neg\beta_n \notin \mathbf{E}$). Tento způsob kontroly konsistence však činí tuto definici nekonstruktivní.

4.4.2 Problémy s rozšířeními pomocí default-negace

Následující příklad ilustruje možnost vytvoření více než jednoho rozšíření teorie.

Příklad 4.3



Vyjádření tvrzení zapsaných v tradiční výrokové logice
rodič_má_dítě & ~adoptované_dítě → plodný_rodič
rodič_má_dítě & ~plodný_rodič → adoptované_dítě
jako default-negace teorie :
T = {rodič_má_dítě}
D = {({rodič_má_dítě}, {adoptované_dítě}, plodný_rodič), ({rodič_má_dítě}, {plodný_rodič}, adoptované_dítě)}

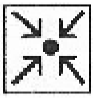
Při výše uvedené interpretaci negace má tato teorie dvě možná rozšíření :
E = {rodič_má_dítě, plodný_rodič}, do něhož nelze zařadit adoptované_dítě
E‘ = {rodič_má_dítě, adoptované_dítě}, do něhož nelze zařadit plodný_rodič.

4.4.3 Default důkazy jako argumenty

Důležitou vlastností default závěrů je to, že nejsou kategorické. Default pravidel se používá ke konstrukci *podmíněných důkazů*, tj. takových, které mohou být na základě dodatečné informace zpochybněny. Takové důkazy mají spíše charakter *argumentů*.

Default důkazy lze, jak ukazuje následující příklad, považovat za vyvratitelné argumenty.

Příklad 4.4



Dr. F očekává příjezd Prof. L do Londýna na plánované setkání, a to jako obvykle vlakem. Jeho sekretářka ale zjistí, že londýnská nádraží budou v předpokládanou dobu mimo provoz, proto argumentuje, že L přijede na setkání se zpožděním. F ale uvažuje takto : Jestliže L stihne dřívější vlak, nemusí přijet pozdě. Tím zpochybní sekretářčin argument. Alternativně L může přijet autobusem, čímž se zpochybní předpoklad jeho příjezdu vlakem a tím i celý argument.

Default pravidlo δ z **D** :

uzavřené_stanice : cestuje_vlakem(x) / zpožděný(x)

Dále jsou zde dvě pravidla z **T** :

r_1 :$\forall$x (stihne_dřívější_vlak(x) $\rightarrow \neg$zpožděný(x))

r_2 :$\forall$x (cestuje_autobusem(x) $\rightarrow \neg$cestuje_vlakem(x))

Protože jde o uzavřenou default teorii, je rovněž pravidlo δ považováno za univerzální.

Default teorie :

Δ = ({uzavřené_stanice, r_1, r_2 },{δ})

r_1, r_2, cestuje_vlakem(L)

r_1, r_2, stihne_dřívější_vlak(L)$\rightarrow\neg$zpožděný(L),
cestuje_autobusem(L)$\rightarrow\neg$cestuje_vlakem(L), cestuje_vlakem(L)

r_1, r_2, st_dřívější_vlak(L)$\rightarrow\neg$zpožděný(L),
$\neg$cestuje_autobusem(L), cestuje_vlakem(L)

r_1, r_2, st_dřívější_vlak(L)$\rightarrow\neg$zpožděný(L),
$\neg$cestuje_vlakem(L), cestuje_vlakem(L)

r_1, r_2, $\neg$st_dřívější_vlak(L),
$\neg$cestuje_autobusem(L),
cestuje_vlakem(L)

r_1, r_2,$\neg$zpožděný(L),
$\neg$cestuje_autobusem(L),
cestuje_vlakem(L)

obr. 4.4

Protože zde není možnost odvození $\neg$cestuje_vlakem(L), bude extenze obsahovat formuli zpožděný(L) získané aplikací pravidla δ.

Nyní předpokládejme, že L stihl dřívější vlak. Máme proto default teorii
$\Delta' = (T', D)$, kde $T' = T \cup \{$stihne_dřívější_vlak(L)$\}$.

Nyní aplikujme pravidlo δ na zpožděný(L), který by již měl být součástí extenze. Extenze musí též obsahovat všechny klasické věty z T‘ a default závěry. Podle pravidla r_1 lze odvodit ¬zpožděný(L). Je-li však zpožděný(L) součástí extenze, je odvozen spor – false, z něhož jako předpokladu lze odvodit cokoliv, např. ¬stihne_dřívější_vlak(L). Odvozené ¬zpožděný(L) tedy vyvrací argument o zpoždění, proto

$$\text{zpožděný(L)} \notin E'$$

Předpokládejme dále, že L cestuje autobusem. Nyní je
$\Delta'' = (T'', D)$, kde $T'' = T \cup \{$cestuje_autobusem(L)$\}$.
S použitím pravidla r_2 odvodíme ¬cestuje_vlakem(L), což opět není konsistentní s cestuje_vlakem(L), proto je argument učiněn neplatným, tedy

$$\text{zpožděný(L)} \notin E''$$

Úkol 4.1

Prověřte, zdali lze závěr odvodit z dané množiny předpokladů **A** pomocí normálního default pravidla :

Jestliže se Jones včera v noci nesetkal se Smithem, pak buď je Smith vrahem nebo Jones lže. Není-li Smith vrahem, Jones se včerejší noci setkal se Smithem a vrah přišel po půlnoci. Přišel-li vrah po půlnoci, je buď Smith vrahem nebo Jones lže. Proto vrahem je Smith.

Default teorie Δ = (**T**, **D**) je dvojice množin, kde **T** je množinou uzavřených formulí logiky prvního řádu a **D** je množina default pravidel.
Default teorie Δ = (**T**, **D**) je *normální*, jsou-li všechna její default pravidla z **D** normální.

Množina důsledků default teorie se nazývá jejím *rozšířením*.

Základní vlastností default závěrů je to, že nejsou kategorické. Default pravidel se používá ke konstrukci *podmíněných důkazů*, které mohou být na základě dodatečné informace zpochybněny. Takové důkazy mají spíše charakter *argumentů*.

Rozšíření teorie, které zůstane konsistentní i po přidání všech odvozených závěrů, se nazývá *regulární*.

Požadavek, aby každá formule z rozšíření byla odvoditelná na základě skutečností tvrzených ve výchozí množině speciálních axiómů, zajišťuje jeho *založenost* (groundedness).

Jestliže byla při budování rozšíření teorie všechna aplikovatelná pravidla aplikována, rozšíření bylo *saturováno*.

Kontrolní otázky

1. Co je default teorie a její rozšíření ?
2. Jaký je rozdíl mezi důkazy a argumenty v logice ?
3. Jaké jsou základní požadavky na rozšíření teorie ?

Literatura

1. Antoniou, G., MacNish, Cara, Foo, N.Y. : A note on the Refinement of Nonmonotonic Knowledge Bases. Knowledge and Information Systems 2. 479 – 486, Springer Verlag London Ltd., 2000.
2. Krause,P , Clark, D : Representing Uncertain Knowledge
   Kluwer Academic Publishers,1993.
3. Polos, L., Hannan, M.T. : Reasoning with Partial Knowledge
   Research Paper No 1638, Stanford University, 2000.
4. Reiter, R. : A logic for default reasoning. Artificial Intelligence 13, 81 – 132, 1980.

5 MODÁLNÍ LOGIKA I

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se seznámíte s další možností, jak zavést do tradiční logiky prvního řádu jistou neurčitost, která by poněkud přiblížila formalizované odvozování v systémech logiky prvního řádu skutečnému usuzování, tak jak probíhá v lidských myslích.

5.1 MODÁLNÍ LOGIKA A LOGIKA PRVNÍHO ŘÁDU

5.1.1 Přístup modální logiky

Pro přístup modální logiky je charakteristické, že se pokouší zachytit taková tvrzení přirozeného jazyka, jejichž pravdivostní hodnota není jednoznačně true nebo false, ale závisí na podmínkách vnějšího světa a jejich změnách v čase. Tato tvrzení zpravidla obsahují slova jako „možná“, „snad“, „asi“, „zpravidla“ apod., časový rozměr pak bývá vyjadřován pomocí „bylo“, „bude“, „je“.

Definice 5.1

Možnost a nutnost se nazývají *aletické modality* nebo *módy pravdivosti*. Logické systémy obsahující operátory pro „je možné, že“ (označuje se $\Diamond$) , „je nutné, aby“ (označuje se $\Box$) se též nazývají *aletické logiky*.

Existuje řada logik, vycházejících z principu zde uváděné modální logiky a zavádějících další operátory.
Deontická logika např. zavádí tyto operátory :
O – „závazně (obligatory) platí…“
P – „je přípustné (permitted), aby platilo…“
F – „je zakázáno, aby platilo…“
Temporální logika disponuje módy pro „někdy“ a „vždy“, spolu s negacemi „nikdy“ a „ne vždy“:
G – „vždy bude platit, že…“
F – „někdy bude platit, že…“
H – „vždy platilo, že…“
P – „někdy platilo, že…“

Příklad 5.1

Příklady reprezentace tvrzení s aletickými módy pravdivosti :

„Je možné, že Jan poslal Marii knihu“ : $\Diamond$ poslat(jan, marie, kniha)

„Je jisté, že vlak má zpoždění a proto Jan dnes nepřijde včas. “ :
$\Box$(zpožděný(vlak) $\rightarrow \neg$přijít(Jan, dnes, včas))
„Není možné, aby Jan poslal něco někomu.“ :

$\neg(\Diamond(\exists x \exists y$ (předmět(x) & osoba(y) & poslat(jan, x, y)))

5.2 JAZYK $\mathbf{L_M}$ VÝROKOVÉ MODÁLNÍ LOGIKY

Jazyk modální logiky je nadmnožinou jazyka $\mathbf{L_1}$. Je třeba, aby byl schopen vyjádřit všechno, co umí logika prvního řádu. Interpretace konstant a formulí s kvantifikátory v sémantice možných světů je problematická, proto se zde omezíme na výrokovou variantu logiky prvního řádu.

5.2.1 Syntax jazyka L_M

Všechna syntaktická pravidla jazyka **L** výrokové logiky jsou zároveň pravidly jazyka $\mathbf{L_M}$ výrokové modální logiky.

<u>Rozšíření jazyka pro vyjádření módů pravdivosti</u> :

- K vyjádření možnosti a nutnosti se zde zavádějí dva modální operátory : ◊ a □ .
  V této symbolice je
  $$\Diamond(P) \Leftrightarrow \neg\Box(\neg P),$$
  $$\text{resp. } \Box(P) \Leftrightarrow \neg\Diamond(\neg P),$$
  proto modální logika vystačí s jedním z nich (zde operátor □).
- Je-li F formule jazyka **L** a M modální operátor, pak M(F) je formule modální logiky.

Definice 5.2 (gramatiky jazyka $\mathbf{L_M}$ Backus-Naurovou formou)

< atom > ::= < výroková proměnná >

::= < logická konstanta >

< formule > ::= < atom >

::= ¬(<formule>)

::= < modální operátor > (<formule>)

::= (< formule >)< binární logická spojka >(< formule >)

< výroková proměnná > ::= a / b / c /......

< logická konstanta > ::= true / false

< binární logická spojka > ::= & / ∨ / → / ↔

< modální operátor > ::= □ / ◊

Příklad 5.2

Formulujte následující tvrzení v jazyce $\mathbf{L_M}$ a výsledné formule přepište s použitím pouze modálního operátoru □.

1. Ptáci létají nízko, proto je možné, že bude pršet.
2. Je teprve únor, proto je pravděpodobné, že se mrazy vrátí.
3. Není jisté, že se akce bude konat už tento měsíc, spíše to bude až příští měsíc.

Řešení :

1. ptáci_létají_nízko $\rightarrow \Diamond$(bude_pršet), ptáci_létají_nízko $\rightarrow \neg\Box(\neg$bude_pršet)
   (V antecedentu se nejedná o zcela jisté tvrzení, ale pouhé konstatování toho, co pozorovatel vidí ve svém okolí)
2. $\Box$(teprve_je_únor) $\rightarrow \Diamond$(mrazy_se_vrátí), $\Box$(teprve_je_únor) $\rightarrow \neg\Box(\neg$mrazy_se_vrátí)
3. $\neg\Box$(akce_bude_tento_měsíc) & $\Diamond$(akce_bude_příští_měsíc),
   $\neg\Box$(akce_bude_tento_měsíc) & $\neg\Box(\neg$akce_bude_příští_měsíc)

5.2.2 Sémantika jazyka L_M

Modální logika dokáže vystihnout rozdíl mezi tím, co je pravda, resp. nemůže být nepravda (bezpodmínečně nutná pravda) a co by mohlo být pravda (podmíněná pravda). Test, o kterou z pravd se jedná, pak spočívá v možné či nemožné představě světa, tj. modelu, v němž je to tak nebo jinak. V modální logice se tedy reprezentovaná tvrzení vztahují současně k několika světům, v nichž tato tvrzení mohou nabývat rozdílných pravdivostních hodnot.

Sémantika jazyka $\mathbf{L_M}$ je definována v pojmech *možných světů* $\{W_0, W_1,\ldots\}$ a *relace přístupnosti* R mezi nimi. Tomu se říká *modální rámec*.

Protože logika prvního řádu pojednává o formulích, které jsou pravdivé v jednotlivých svých modelech - světech, jeví se přirozené uvažovat modely logiky prvního řádu jako možné světy $\{W_0, W_1,\ldots\}$ pro modální logiku.

5.2.3 Pojem možných světů a jeho sémantiky

Definice 5.3 (možného světa)

Možným světem dané množiny formulí modální logiky je každý její model, tj. každá struktura pravdivostních hodnot jejích atomů, pro niž je množina formulí interpretována jako true.

Definice 5.4 (modálního rámce)

Modálním rámcem je dvojice ($\{W_0, W_1,\ldots\}$, R), kde $\{W_0, W_1,\ldots\}$ je množina možných světů, R je relace přístupnosti mezi nimi.

Definice 5.5 (pravdivosti formule modální logiky)

1. Formule tvaru $\Box$(P) je pravdivá ve světě W_i v modálním rámci ($\{W_0, W_1,\ldots\}$, R), právě když P je pravdivá ve všech světech W_j tohoto rámce takových, že W_j je přístupný z W_i, tj. R(W_i, W_j).
2. Každá jiná formule je pravdivá ve světě W_i v rámci ($\{W_0, W_1,\ldots\}$,R), právě když W_i je její model ve smyslu logiky prvního řádu.
3. Formule je pravdivá v daném rámci, je-li pravdivá v každém ze světů tohoto rámce.

Rozšíření pojmu na logickou platnost v obdobném smyslu jako u logiky prvního řádu zde v podstatě není možné, neboť pravdivost se vždy vztahuje k nějakému rámci s určitou relací přístupnosti a nelze proto uvažovat všechny možné rámce. Platnost proto musí být definována vzhledem k vlastnostem relace přístupnosti. Čím je axióm silnější, tím jsou větší omezení na relaci přístupnosti.

5.3 FORMÁLNÍ SYSTÉM VÝROKOVÉ MODÁLNÍ LOGIKY

Axiómatické systémy modální logiky jsou vždy modifikací některého axiómatického systému logiky prvního řádu. Všechny axiómy příslušného systému logiky prvního řádu proto jsou též modálně platné bez ohledu na relaci přístupnosti R. Světy, které tvoří rámec jsou totiž, stejně jako jiné světy, modely těchto axiómů. To je přesně ta vlastnost, kterou potřebujeme pro pravdivost $\Box(A)$ axiómů ve světě W_i.

5.3.1 Axiómy modálních logik

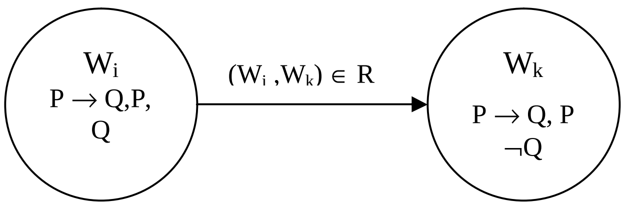
Následující axiómy jsou platné v rámci omezení na relaci R. Přidáním axiómu **K** vzniká **K**-logika apod.

***Axióm* K** $\Box(P \rightarrow Q) \rightarrow (\Box(P) \rightarrow \Box(Q))$

Důkaz (nepřímý) platnosti axiómu **K** :

Abychom mohli uvažovat podmínky na relaci přístupnosti, pro něž je tento axióm platný, musíme uvažovat případy, pro něž neplatí.

Sémantika prvního řádu predikátové/výrokové logiky zná pouze jediný takový případ pro spojku $\rightarrow$: $\Box(P \rightarrow Q)$ i $\Box(P)$ jsou true ve světě W_i a současně je v tomto světě $\Box(Q)$ false. To ale platí, právě když $P \rightarrow Q$ a zároveň P jsou true ve všech světech přístupných z W_i , ale Q je v některém z těchto světů false. Protože všechny tyto světy jsou modely logiky prvního řádu (pravidlo MP „Modus Ponens“ platí), nemůže tento případ nastat.



obr. 5.1

K platí bez ohledu na to, jaká je relace R přístupnosti. Modální logika, která má axióm **K**, se nazývá *normální modální logika*.

Další axióm říká, že každá formule, která je nutně pravdivá, je pravdivá.

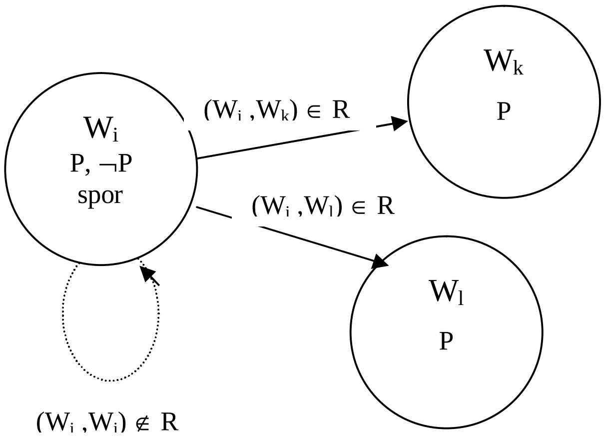
***Axióm* T** :

T : $\Box(P) \rightarrow P$

Zdálo by se, že není nutné zavádět takový axióm. Tato nutnost však vyplývá z pojmu modálního rámce. Uvažujme jako předtím jediný případ, kdy může nastat případ false : P by musela být true ve všech světech přístupných z některého W_i , nikoliv však z tohoto W_i . W_i

by ale musel nebýt přístupný ze sebe sama. Uvažujme, kdy to prakticky může nastat : nechť je relací přístupnosti relace „později než“. Tím se □P interpretuje jako „ stane se pravdivým a zůstane pravdivým“. Pro tento případ platí axióm **K**, nikoliv však axióm **T**. Protipříklad : „Bude pravda a zůstane pravda, že jsem mrtvý“ je pravda, ale „Jsem mrtvý“ není pravda.

T platí ve všech případech, kdy je relace přístupnosti reflexivní.



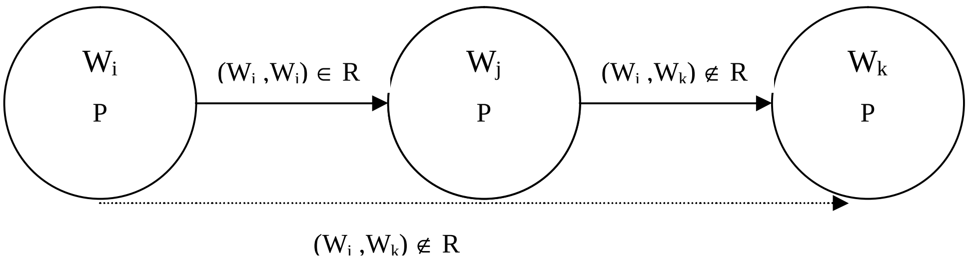
obr. 5.2

Další axióm říká, že když je formule nutně true, pak je nutně nutně true

***Axióm* S4** :

S4 : □(P) → □(□(P))

Opět sledujeme možnost, kdy to není pravda. To může nastat pouze tehdy, jestliže existuje svět W_j přístupný z W_i, ale takový, že v něm □(P) je false. To je možné, je-li zde další svět W_k, který je přístupný z W_j , pro nějž je P false. To ale je možné pouze když W_k není přístupný z W_i. To znamená, že aby neplatil **S4**, nesmí být relace přístupnosti tranzitivní.



obr. 5.3

Příklad 5.3

Uvažujme relaci „Je do 24 hodin.“. Tato relace je jasně reflexivní, ale nikoliv tranzitivní. Relace přístupnosti zde podporuje interpretaci □(P) ve smyslu „bude pravda od tohoto okamžiku do 24 hodin.“

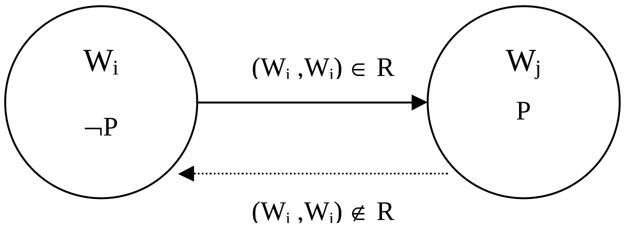
□(datum je 20. nebo 21. června) je pravda, ale není pravda □(□(datum je 20. nebo 21. června))

Axióm **B** :

B : $\neg P \rightarrow \Box(\neg\Box(P))$

Tento axióm říká, že je-li P false, pak je to ten případ, že není nutně true, jinak řečeno že je zde možnost, aby byl false.

Aby axióm **B** neplatil, je třeba, aby existoval v daném rámci svět W_i, v němž by $\neg P$ byla true a zároveň $\neg(\Box(P))$ false. To ale znamená, že P musí být false ve W_i a $\Box(P)$ musí být true v nějakém W_j. To však lze pouze tehdy, není-li W_i přístupný z W_j, neboť kdyby byl, musela by tam být true i formule P. Proto, aby axióm **B** selhal, je třeba, aby existoval pár světů (W_i , W_j) takový, že platí $R(W_i, W_j)$, ale neplatí $R(W_j , W_i)$, tj. relace přístupnosti nesmí být symetrická.



obr. 5.4

5.3.2 Odvozovací pravidla modální logiky

Formální systém výrokové modální logiky, založený na axiómatickém systému **H** hilbertovského typu, disponuje známým odvozovacím pravidlem **MP** (Modus Ponens), a nově zavedeným modálním pravidlem nutnosti **NEC** :

Definice 5.6 (odvozovacího pravidla MP)

Pravidlo **MP** tvaru

$$\frac{\mathbf{U} \mid\!- A \qquad \mathbf{U} \mid\!- A \rightarrow B}{\mathbf{U} \mid\!- B} \quad ,$$

„Z dokazatelnosti vět (formulí) A a $A \rightarrow B$ odvoď větu (formuli) B."

se nazývá *odvozovací pravidlo Modus Ponens.*

Definice 5.7 (odvozovacího pravidla NEC)

Pravidlo **NEC** tvaru

$$\frac{\mathbf{U} \mid\!- A}{\mathbf{U} \mid\!- \Box(A)}$$

„Je-li z **U** dokázána A, pak z ní odvoď □(A)."

se nazývá *odvozovací pravidlo nutnosti* (*necessity*).

5.3.3 Sémantická korektnost modální logiky

Věta 5.1

Modální logika je sémanticky korektní formální systém

Důkaz

Předpokládejme, že F je modální rámec, tj. množina $\{W_0, W_1 \ldots\}$ modelů logiky prvního řádu spolu s relací přístupnosti R, pro nějaký soubor modálních axiómů, který splňuje určité podmínky.

Jestliže existuje důkaz formule *A* z axiómů logiky prvního řádu a tohoto souboru modálních axiómů s použitím MP a NEC, potom *A* platí v rámci F.

Důkaz indukcí podle délky důkazu (využívající sporu s předpokladem, že existuje kratší důkaz, pro nějž věta může selhat) :

Předpokládejme, že *A* je věta, jejíž nejkratší důkaz je tak krátký jako nějaký důkaz, pro nějž věta neplatí. Je třeba uvažovat tři případy :

1) *A* je axióm. Už byla dokázána korespondence mezi axiómy modální logiky a relací přístupnosti – každý axióm je platný ve vhodném rámci
2) *A* vyplývá na základě MP z dokázané formule, tj. z dokázaných *B* a $B \rightarrow A$, pro něž platí indukční předpoklad, se odvodí *A*. Protože každý svět z F je modelem logiky prvního řádu, *A* musí platit v každém z nich, což odporuje předpokladu, že pro něj formule selhává.
3) *A* vyplývá na základě NEC z dokázané formule, pro niž platí indukční předpoklad. Předpokládejme *A* je ve skutečnosti $\Box(B)$, pro niž je dokázáno *B*. Má-li věta *A* selhat, musí v F existovat svět W_i, v němž $\Box(B)$ není true, tedy musí existovat svět W_j přístupný z W_i takový, že v něm je *B* false. To ale odporuje indukčnímu předpokladu.

Existuje mnoho způsobů, jak stanovit axiómy modální logiky.

Úkol 5.1

Formulujte v jazyce tato tvrzení :

Nemohu najít klíče. Je možné, že jsem je ztratil. Dnes ráno jsem určitě měl klíče v kapse. Nemohu najít nejen klíče, ale ani brýle, takže nemohu vyloučit, že jsem poněkud roztržitý.

Modální logika se pokouší zachytit taková tvrzení přirozeného jazyka, jejichž pravdivostní hodnota není jednoznačně true nebo false, ale závisí na podmínkách vnějšího světa a jejich změnách v čase. Tato tvrzení zpravidla obsahují slova jako „možná“, „snad“, „asi“, „zpravidla“ apod., časový rozměr pak bývá vyjadřován pomocí „bylo“, „bude“, „je“.

Možnost a nutnost se nazývají aletické modality nebo módy pravdivosti. Logické systémy obsahující operátory pro „je možné, že“ (označuje se $\Diamond$) , „je nutné, aby“ (označuje se $\Box$) se též nazývají aletické logiky.

Jazyk modální logiky je nadmnožinou jazyka $\mathbf{L_1}$. Všechna syntaktická pravidla jazyka **L** výrokové logiky jsou zároveň pravidly jazyka $\mathbf{L_M}$ výrokové modální logiky.

Rozšíření jazyka pro vyjádření módů pravdivosti :

- K vyjádření možnosti a nutnosti se zde zavádějí dva modální operátory : $\Diamond$ a $\Box$.

V této symbolice je $\Diamond(P) \Leftrightarrow \neg\Box(\neg P)$, resp. $\Box(P) \Leftrightarrow \neg\Diamond(\neg P)$,
proto modální logika vystačí s jedním z nich (zde operátor $\Box$).

- Je-li F formule jazyka L a M modální operátor, pak M(F) je formule modální logiky.

Sémantika jazyka $\mathbf{L_M}$ je definována v pojmech *možných světů* $\{W_0, W_1,\ldots\}$ a *relace přístupnosti* R mezi nimi. Tomu se říká *modální rámec*.

Modálním rámcem je dvojice $(\{W_0, W_1,\ldots\}, R)$, kde $\{W_0, W_1,\ldots\}$ je množina možných světů, R je relace přístupnosti mezi nimi.

Formule tvaru $\Box(P)$ je pravdivá ve světě W_i v modálním rámci $(\{W_0, W_1,\ldots\}, R)$, právě když P je pravdivá ve všech světech W_j tohoto rámce takových, že W_j je přístupný z W_i, tj. $R(W_i, W_j)$.

Každá jiná formule je pravdivá ve světě W_i v rámci $(\{W_0, W_1,\ldots\},R)$, právě když W_i je její model ve smyslu logiky prvního řádu.

Formule je pravdivá v daném rámci, je-li pravdivá v každém ze světů tohoto rámce.

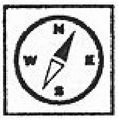
Axiómatické systémy modální logiky jsou vždy modifikací některého axiómatického systému logiky prvního řádu. Všechny axiómy příslušného systému logiky prvního řádu proto jsou též modálně platné bez ohledu na relaci přístupnosti R. Světy, které tvoří rámec jsou, stejně jako jiné světy, modely těchto axiómů.

Formální systém výrokové modální logiky, založený na axiómatickém systému **H** hilbertovského typu, disponuje známým odvozovacím pravidlem **MP** (Modus Ponens)
a nově zavedeným modálním pravidlem nutnosti **NEC** : Je-li $\mathbf{U} \vdash A$, pak je $\mathbf{U} \vdash \Box(A)$. Modální logika je sémanticky korektní formální systém.

Kontrolní otázky

1. V čem spočívá rozšíření jazyka modální logiky vzhledem k logice prvního řádu ?
2. Co je modální rámec ?
3. Kdy je pravdivá formule $\Box(P)$ ve světě W_i v modálním rámci $(\{W_0, W_1,\ldots\}, R)$?
4. Kdy je pravdivá formule $\Box(P)$ v modálním rámci $(\{W_0, W_1,\ldots\}, R)$?
5. Jak je vybudován formální systém výrokové modální logiky ?

6 MODÁLNÍ LOGIKA II



V této lekci se v návaznosti na lekci předcházející se seznámíte s tablovou metodou důkazu platnosti formulí jazyka $\mathbf{L_M}$. *Výroková verze modální logiky pak bude doplněna ukázkami reprezentace v predikátové verzi modální logiky. Prostudování této lekce by vám mělo trvat zhruba 1,5 h.*

6.1 SÉMANTICKÉ TABLO FORMULE VÝROKOVÉ MODÁLNÍ LOGIKY

Formuli $\Box(P)$ uvažujeme ve smyslu $\forall w\ P(w)$, tj. zavádíme zvláštní argument w označující možné světy, v nichž formule platí.

Při tablové důkazové metodě je třeba sledovat stopu světů reprezentovaných určitými valuacemi, abychom byli schopni testovat přístupnost z jednoho do druhého. Způsob realizace tohoto sledování spočívá v přičlenění indexu ke každé formuli, jehož smyslem je vytvářet omezení na množině světů, v nichž má být formule true.

6.1.1 Indexování možných světů

Definice 6.1

Indexem může být proměnná, konstanta nebo term tvaru succ(I_1, I_2). Poslední z nich určuje cestu, kde I_2 je výchozí svět, I_1 je místo určení. Konstanty se značí malým, proměnné velkým písmenem.

Definice 6.2

Indexovaná formule je dvojice F:I sestávající z formule jazyka $\mathbf{L_M}$ výrokové varianty modální logiky a indexu.
Dvě indexované formule F_1:I_1 a F_2:I_2 jsou komplementární, jestliže
1) F_1 je negací F_2 a naopak
2) I_1 a I_2 jsou unifikovatelné.

Definice 6.3

Modální tablo je binární strom ohodnocený indexovanými formulemi.

6.1.2 Tablová odvozovací pravidla

Definice 6.4

Nechť Φ je množina formulí, P a Q jsou formule jazyka $\mathbf{L_M}$. Potom platí následující non-modální tablová pravidla (odpovídající tablovým pravidlům logiky prvního řádu) :
1) α-pravidla :
   Je-li $\Phi \cup \{(P\&Q):I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští

$\Phi \cup \{(P\&Q):I, P:I, Q:I\}$.
Je-li $\Phi \cup \{\neg(P\vee Q):I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{\neg(P\vee Q):I, \neg P:I, \neg Q:I\}$.
Je-li $\Phi \cup \{\neg(P\rightarrow Q):I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{\neg(P\rightarrow Q):I, P:I, \neg Q:I\}$.

2) β-pravidla :
Je-li $\Phi \cup \{(P\vee Q):I\}$ návěští uzlu modálního tabla, potom uzly dvou následujících větví mají návěští $\Phi \cup \{(P\vee Q):I, P:I\}$, $\Phi \cup \{P\vee Q:I, Q:I\}$,
Je-li $\Phi \cup \{\neg(P\&Q):I\}$ návěští uzlu modálního tabla, potom uzly dvou následujících větví mají návěští $\Phi \cup \{\neg(P\&Q):I, \neg P:I\}$, $\Phi \cup \{P\vee Q:I, \neg Q:I\}$,
Je-li $\Phi \cup \{(P\rightarrow Q):I\}$ návěští uzlu modálního tabla, potom uzly dvou následujících větví mají návěští $\Phi \cup \{(P\rightarrow Q):I, \neg P:I\}$, $\Phi \cup \{P\vee Q:I, Q:I\}$,

Definice 6.5

Nechť Φ je množina formulí jazyka $\mathbf{L_M}$. Potom platí následující modální tablová pravidla :

3) Je-li $\Phi \cup \{\neg(\Box(P)):I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{\neg(\Box(P)):I, \neg P:succ(I', I)\}$, kde I' je nová konstanta.
4) Je-li $\Phi \cup \{\Box(P):I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{\Box(P):I, P:succ(I', I)\}$, kde I' je nová proměnná.

6.1.3 Příklady modálních tablových důkazů

Příklad 6.1

Nepřímý tablový důkaz $\Box(P\&Q) \rightarrow \Box(P)$:

$$\neg(\Box(P\&Q) \rightarrow \Box(P)):w_0$$
$$|$$
$$\Box(P\&Q):w_0,\ \neg\Box(P):w_0$$
$$|$$
$$\Box(P\&Q):w_0,\ (P\&Q):succ(W_0, w_0),\ \neg\Box(P):w_0$$
$$|$$
$$\Box(P\&Q):w_0,\ (P\&Q):succ(W_0,w_0),\ P:succ(W_1,w_0),\ Q:succ(W_2,w_0), \neg\Box(P):w_0, \neg P:succ(w_1,w_0)$$
$$\times$$

obr. 6.1

Tablo se uzavře, neboť je zde možná unifikace substitucí $\{w_1/W_1\}$.

Příklad 6.2

Nepřímý tablový důkaz axiómu **K** :

$$\neg(\Box(P \rightarrow Q) \rightarrow (\Box(P) \rightarrow \Box(Q)):w_0$$

$$\Box(P \rightarrow Q):w_0,\ \neg(\Box(P) \rightarrow \Box(Q)):w_0$$

$$\Box(P \rightarrow Q):w_0,\ \Box(P):w_0,\ \neg\Box(Q):w_0$$

$$\Box(P \rightarrow Q):w_0,\ \Box(P):w_0,\ P:succ(W,\ w_0),\ \neg\Box(Q):w_0,\ \neg Q:succ\ (w_1,\ w_0),\ (P \rightarrow Q):\ succ(W',\ w_0)$$

$\Box(P \rightarrow Q):w_0,\ \Box(P):w_0,\ P:succ(W,\ w_0),$ $\neg\Box(Q):w_0,\ \neg Q:succ\ (w_1,\ w_0),$ $\neg(P):\ succ(W',\ w_0)$	$\Box(P \rightarrow Q):w_0,\ \Box(P):w_0,\ P:succ(W,\ w_0),$ $\neg\Box(Q):w_0,\ \neg Q:succ\ (w_1,\ w_0),$ $(Q):\ succ(W',\ w_0)$
×	×

obr. 6.2

Tablo se uzavře, neboť je zde možná unifikace substitucí { $w_1/W,\ w_1/W'$}.

Příklad 6.3

Nepřímý tablový důkaz $(\Box P \vee \Box Q) \rightarrow \Box(P \vee Q)$

$$\neg(\Box P \vee \Box Q) \rightarrow \Box(P \vee Q):w_0$$

$$(\Box P \vee \Box Q):w_0,\ \neg\Box(P \vee Q):w_0$$

$$(\Box P \vee \Box Q):w_0,\ \neg\Box(P \vee Q):w_0, \neg(P \vee Q):succ(w_1,w_0)$$

$(\Box P\vee\Box Q):w_0,\ \neg\Box(P\vee Q):w_0$ $\neg(P\vee Q):succ(w_1,w_0),\ \Box(P):w_0, P:w_0$ $\neg P:succ(w_1,w_0),\ \neg Q:succ(w_1,w_0),$ $P:succ(W_0,w_0)$	$(\Box P\vee\Box Q):w_0,\ \neg\Box(P\vee Q):w_0$ $\neg(P\vee Q):succ(w_1,w_0),\ \Box(Q):w_0, Q:w_0$ $\neg P:succ(w_1,w_0),\ \neg Q:succ(w_1,w_0),$ $Q:succ(W_0,w_0)$
×	×

obr. 6.3

Tablo se uzavře, neboť je zde možná unifikace substitucí { w_1/W_0}.

Příklad 6.4

Důkaz, že neplatí $\Box P \rightarrow P$:

$$\neg(\Box P \rightarrow P):w_0$$

$$\Box(P):w_0,\ P:succ(W_0,w_0),\ \neg P:w_0,$$

○

obr. 6.4

Tablo se neuzavře, proto $\Box P \rightarrow P$ neplatí.

Příklad 6.5

Důkaz, že neplatí $\Box(P \vee Q) \rightarrow (\Box P \vee \Box Q)$

$$\neg(\Box(P \vee Q) \rightarrow (\Box P \vee \Box Q)):w_0$$

$$\Box(P \vee Q):w_0,\ \neg(\Box P \vee \Box Q)):w_0$$

$$\Box(P \vee Q):w_0,\ \neg\Box(P):w_0,\ \neg\Box(Q):w_0$$

$$\Box(P\vee Q):w_0,\ (P\vee Q):succ(W_0,w_0),\ \neg\Box(P):w_0, \neg P:succ(w_1,w_0), \neg\Box(Q):w_0, \neg Q:succ(w_1,w_0)$$

$$\Box(P\vee Q):w_0,\ P:succ(W_0,w_0),\ldots \quad \neg P:succ(w_1,w_0),\ \neg Q:succ(w_1,w_0) \qquad \Box(P\vee Q):w_0,\ Q:succ(W_0,w_0),\ldots \quad \neg P:succ(w_1,w_0),\ \neg Q:succ(w_2,w_0)$$

○ ○

obr. 6.5

Zdálo by se, že se obě větve uzavírají, jestliže se provede unifikace substitucí w_1 za proměnnou W_0 v případě levé větve a substitucí w_2 za proměnnou W_0 v případě pravé větve. To však jsou vzájemně nekompatibilní vazby, proto nepřímý důkaz logické platnosti této formule selhává.

Protipříklad : $\Box(P\vee\neg P)$ platí podle NEC pro libovolné P, ale nelze z toho vyvodit, že platí $\Box(P) \vee \Box(\neg P)$.

Ve výrokové verzi modální logiky, v níž platí axióm K, tablová metoda v každém případě rozhoduje, podobně jako je tomu ve výrokové logice, splnitelnost formule (taktéž logický důsledek dané množiny predpokladů). Pro její rozšíření na další typy modální logiky je třeba uvažovat alternativní pravidla, zavádějící indexy potenciální kontradikce. Index formule odráží pohyb, který bereme v úvahu, prokazujeme-li existenci světa, v němž formule platí.

Pro důkazy axiómů T a S4 je třeba uvažovat vlastnosti indexů vyplývající z reflexivnosti a tranzitivnosti relace přístupnosti počítat s vymazáváním pohybů z cesty.

Příklad 6.6

Důkaz axiómu T : $(\Box P \rightarrow P)$

$$\neg(\Box P \rightarrow P):w_0$$

$$|$$

$$\Box(P):w_0,\ \neg P:w_0,\ P:succ(W_0,w_0)$$

$$\times$$

obr. 6.6

Vzhledem k relaci R_T jde o akceptovatelný důkaz, neboť R_T $(w_0, succ(W_0,w_0))$ v této relaci, která je reflexivní, platí.

Příklad 6.7

Důkaz S4 $(\Box P \rightarrow \Box(\Box P))$:

$$\neg(\Box P \rightarrow \Box(\Box P)):w_0$$

$$|$$

$$\Box(P):w_0,\ \neg(\Box(\Box P)):w_0,\ \neg\Box P:succ(w_1,w_0),\ \neg P:succ(w_2,\ succ(w_1,w_0)),\ P:succ(W_0,w_0)$$

$$\times$$

obr. 6.7

Vzhledem k relaci přístupnosti R_{S4} , která je tranzitivní, je to akceptovatelný důkaz, neboť R_{S4} $(succ(w_2, succ(w_1,w_0)), succ(W_0,w_0))$ platí.

6.2 MODÁLNÍ LOGIKA PRVNÍHO ŘÁDU

6.2.1 Pevný designátor

V případě predikátové verze modální logiky je stejně jako u tradiční predikátové logiky při důkazech zpravidla nutno provádět unifikaci. Existenční kvantifikace musí být spojena se skolemizací, která odkazuje na indexy světů, v nichž má být formule pravdivá. K tomu slouží náhrada pravidla pro formule tvaru $\Box(P):I$ takto :

5) Je-li $\Phi \cup \{\Box(P):I\}$ návěští uzlu modálního tabla, potom návěští následujícího uzlu lze rozšířit
   $\Phi \cup \{\Box(P):I, P:I\}$, je-li místem určení z I proměnná, a
   $\Phi \cup \{\Box(P):I, P:succ(V,I)\}$, kde V je nová proměnná, je-li místem určení z I konstanta.

Problém identity individuí v rámci možných světů : V případě logiky prvního řádu máme množinu konstant (jmen), jimž při konstrukci modelu jazyka přiřazujeme denotáty – objekty z W. Podobně je tomu v případě funkcí. Při úvahách o možných světech však tento způsob denotace nepřipadá v úvahu.. Musíme naopak rozhodovat, která jména konstantních jazykových symbolů jsou spolehlivými *pevnými designátory*, resp. *pevnými funkcemi* v mezích možných světů.

Specializace pevný designátor, resp. pevná funkce je další komponentou definice relace přístupnosti. Je-li např. matka pevným predikátem v nějakém světě, pak pouze ty světy, které mají vlastnost, že určitá osoba je označena jako moje matka, je právě moje matka, zachovávají, jsou z uvažovaného světa přístupné. To vede k platnosti Barcanových formulí.

6.2.2 Barcanovy formule :

B1 : $\Box(\forall x\, P(x)) \leftrightarrow \forall x\, (\Box P(x))$

Tvrdíme-li, že nutně P(x) platí pro všechna x, pak je to totéž, jako když tvrdíme, že pro všechna x nutně platí P(x). $\forall x\, (\Box P(x))$ znamená, že pro všechny prvky daného světa je P true pro tyto prvky ve všech přístupných světech. Všechny světy přístupné z uvažovaného mají stejné prvky, i když se jejich jména mohou lišit.

B2 : $\exists x\, (\Box P(x)) \leftrightarrow \Box(\exists x P(x))$

Tvrdíme-li, že existuje x, pro něž nutně platí P(x), pak je to totéž, jako když tvrdíme, že nutně existuje x tak, že platí P(x). V terminologii možných světů to znamená : $\exists x\, (\Box P(x))$ říká, že existuje prvek uvažovaného světa, pro nějž P platí ve všech z něj přístupných světech. To kontrastuje s $\Box(\exists x P(x))$, která říká, že v kterémkoliv přístupném světě existuje prvek, pro nějž platí P.

Předpokládejme predikát matka(x, Julian). Potom $\Box(\exists x$ matka(x, Julian)) vyžaduje, abychom se zabývali pouze přístupnými světy, v nichž má Julian matku. Pouze když uvažujeme matku jako pevný predikát, můžeme akceptovat $\exists x\, (\Box$matka(x, Julian)), že je zde někdo, kdo je v každém možném světě Julianovou matkou.

Akceptujeme-li nebo nikoliv B1, B2, má to své důsledky pro relaci přístupnosti. B1 se jeví jako platný, právě když přístupné světy mají tytéž prvky. B2 platí pro pevné vlastnosti, tj. cokoliv dosadíme za x je pevný designátor a ne pouze jméno, jehož hodnota se mění od světa ke světu. Výběr, které vlastnosti reprezentované predikáty jsou pevné a které ne, je hlavním problémem relace přístupnosti. Má to velký význam z hlediska vhodnosti modální logiky pro umělou inteligenci.

6.2.3 Modální tablové důkazy prvního řádu

$\Box(P)$ je ekvivalentní formuli $\forall W(R(W,w_0) \rightarrow P)$. $\Box(P)$ platí v reálném světě w_0, právě když P platí ve všech světech, které jsou v relaci s w_0.

Pro adaptaci tablové metody na modální logiku je třeba adaptovat konstrukci indexů. Ukážeme to zde na důkazu $\exists X\Box(P) \rightarrow \Box(\exists XP)$. Ukážeme též, že neplatí $\Box(\exists XP) \rightarrow \exists X\Box(P)$. K tomu ale potřebujeme nová pravidla pro kvantifikátory.

6) Je-li $\Phi \cup \{\neg\forall X\ P{:}I\}$ návěští uzlu modálního tabla, potom návěští následujícího uzlu lze rozšířit

$\Phi \cup \{\neg\forall X\ P{:}I,\ \exists X\ \neg P{:}I\}$
Je-li $\Phi \cup \{\neg\exists X\ P{:}I\}$ návěští uzlu modálního tabla, potom návěští následujícího uzlu lze rozšířit
$\Phi \cup \{\neg\exists X\ P{:}I,\ \forall X\ \neg P{:}I\}$.

7) Je-li $\Phi \cup \{\exists X\ P{:}I\}$ návěští uzlu modálního tabla, potom návěští následujícího uzlu lze rozšířit
$\Phi \cup \{\exists X\ P{:}I,\ P|_{<X,\ sk(D)>}{:}I\ \}$, kde sk je nová Skolemova fce a D je destinace I.

8) Je-li $\Phi \cup \{\forall X\ P{:}I\}$ návěští uzlu modálního tabla, potom návěští následujícího uzlu lze rozšířit
$\Phi \cup \{\forall X\ P{:}I,\ P|_{<X,\ a)>}{:}I\ \}$, kde a je konstanta, která se již někde v tablu vyskytla. Je-li zvolenou konstantou Skolemova fce, jejímž jedním argumentem je destinace, kterou je proměnná, musí tato proměnná být unifikována s I.

Příklad 6.8

a) Důkaz $\exists X\Box(P(X)) \rightarrow \Box(\exists XP(X))$.

$$\neg(\exists X\Box(P(X)) \rightarrow \Box(\exists XP(X))){:}w_0$$
$$|$$
$$\exists X\Box(P(X)){:}w_0,\ \neg\Box(\exists XP(X)){:}w_0,\ \neg\exists XP(X){:}succ(w_1,w_0)$$
$$|$$
$$\exists X\Box(P){:}w_0,\ \neg\Box(\exists XP(X)){:}w_0,\ \forall X\neg P(X){:}succ(w_1,w_0)$$
$$|$$
$$\exists X\Box(P){:}w_0,\ \Box(P(sk0,w_0)){:}w_0,\neg\ \Box(\exists XP(X)){:}w_0,\forall X\neg P(X){:}succ(w_1,w_0)$$
$$|$$
$$\exists X\Box(P){:}w_0,\ \Box(P(sk0,w_0)){:}w_0,P(sk0,w_0)){:}succ(W_0,w_0),\neg\ \Box(\exists XP(X)){:}w_0,$$
$$\forall X\neg P(X){:}succ(w_1,w_0),\ \neg P(sk0,w_0){:}succ(w_1,w_0)$$
$$\times$$

obr. 6.8

b) Nepřímý důkaz, že $\Box(\exists X\ P(X)) \rightarrow \exists X\Box(P(X))$ neplatí:

$$\neg(\Box(\exists XP(X)) \rightarrow \exists X\Box(P(X))){:}w_0$$
$$|$$
$$\Box(\exists XP(X)){:}w_0,\neg\exists X\Box(P(X)){:}w_0,$$
$$|$$
$$\Box(\exists XP(X)){:}w_0,\ \exists XP(X)){:}succ(\_W,W_0),\forall X\Box(P(X)){:}w_0,$$
$$|$$
$$\bigcirc$$

obr. 6.9

Modální tablo je binární strom ohodnocený indexovanými formulemi.
Indexem může být proměnná, konstanta nebo term tvaru succ(I_1, I_2). Poslední z nich určuje cestu, kde I_2 je výchozí svět, I_1 je místo určení. Konstanty se značí malým, proměnné velkým písmenem.
Indexovaná formule je dvojice F:I sestávající z formule jazyka $\mathbf{L_M}$ výrokové varianty modální logiky a indexu.
Dvě indexované formule $F_1{:}I_1$ a $F_2{:}I_2$ jsou komplementární, jestliže
1. F_1 je negací F_2 a naopak
2. I_1 a I_2 jsou unifikovatelné.

Tablová odvozovací pravidla :
Nechť Φ je množina formulí, P a Q jsou formule jazyka $\mathbf{L_M}$. Potom platí následující non-modální tablová pravidla (odpovídající tablovým pravidlům logiky prvního řádu) :

1. α-pravidla :
   Je-li $\Phi \cup \{(P\&Q){:}I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{(P\&Q){:}I, P{:}I, Q{:}I\}$.
   Je-li $\Phi \cup \{\neg(P\vee Q){:}I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{\neg(P\vee Q){:}I, \neg P{:}I, \neg Q{:}I\}$.
   Je-li $\Phi \cup \{\neg(P\rightarrow Q){:}I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští $\Phi \cup \{\neg(P\rightarrow Q){:}I, P{:}I, \neg Q{:}I\}$.
2. β-pravidla :
   Je-li $\Phi \cup \{(P\vee Q){:}I\}$ návěští uzlu modálního tabla, potom uzly dvou následujících větví mají návěští $\Phi \cup \{(P\vee Q){:}I, P{:}I\}$, $\Phi \cup \{P\vee Q{:}I, Q{:}I\}$,
   Je-li $\Phi \cup \{\neg(P\&Q){:}I\}$ návěští uzlu modálního tabla, potom uzly dvou následujících větví mají návěští $\Phi \cup \{\neg(P\&Q){:}I, \neg P{:}I\}$, $\Phi \cup \{P\vee Q{:}I, \neg Q{:}I\}$,
   Je-li $\Phi \cup \{(P\rightarrow Q){:}I\}$ návěští uzlu modálního tabla, potom uzly dvou následujících větví mají návěští $\Phi \cup \{(P\rightarrow Q){:}I, \neg P{:}I\}$, $\Phi \cup \{P\vee Q{:}I, Q{:}I\}$,

Nechť Φ je množina formulí jazyka $\mathbf{L_M}$. Potom platí následující modální tablová pravidla :

3. Je-li $\Phi \cup \{\neg(\Box(P)){:}I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští
   $\Phi \cup \{\neg(\Box(P)){:}I, \neg P{:}succ(I',I)\}$, kde I‘ je nová konstanta.
4. Je-li $\Phi \cup \{\Box(P){:}I\}$ návěští uzlu modálního tabla, potom následující uzel má návěští
   $\Phi \cup \{\Box(P){:}I, P{:}succ(I',I)\}$, kde I‘ je nová proměnná.

Kontrolní otázky

1. V čem je rozdíl mezi sémantickým tablem tradiční výrokové logiky a modální výrokové logiky ?
2. Jak se v modálních důkazech indexují možné světy ?

Literatura

Hughes, G.E., Cresswell, M.J. : A new introduction to modal mogic.
Routledge, London, 1996.

7 TERMINOLOGICKÉ (KONCEPTOVÉ) SYSTÉMY REPREZENTACE ZNALOSTÍ

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1 h, se dozvíte, že ve snaze o vytváření takových znalostních bází, s jejichž znalostmi by ve stejné kvalitě, jaké docílí člověk, byl schopen zacházet i automat (softwarový agent) byl vymyšlen nový přístup k reprezentaci znalostí, založený na konceptovém principu a jednoznačně zavedené terminologii.

7.1 ZDROJE TERMINOLOGICKÉ REPREZENTACE ZNALOSTÍ

Konceptový (terminologický) přístup k reprezentaci znalostí, který vyústil v *konceptové jazyky*, prošel v nedávné minulosti jistými vývojovými fázemi, v nichž podstatnou úlohu sehrály zejména :

- Asociativní sítě
- Rámce
- Predikátová logika
- Objektová reprezentace znalostí

7.1.1 Asociativní sítě

V případě asociativních sítí, jak je vidět v lekcích 1,2, jde o modelovací nástroj – jazyk s vysokým stupněm srozumitelnosti a při dodržení určitých jejích pravidel s možností přepisu do fragmentu predikátové (klauzulární) logiky. Stejně jako v predikátové logice vychází reprezentace asociativními sítěmi z atomů vhodně zvolených predikátů. Na rozdíl od logiky 1. řádu však asociativní síť disponuje pouze dvoumístnými predikáty, pouze takové relace lze totiž reprezentovat jako síť, která má své hrany a uzly. Hrany nesou jako svá návěští predikátové symboly, uzly symboly přípustné v predikátových atributech, tj. termy pojmenovávající objekty reprezentovaného světa.

Problémy s asociativními sítěmi souvisí s jejich nejasnou sémantikou, způsobenou především zařazováním do společných sítí tvrzení o obecných třídách (kategoriích) s tvrzeními, týkajícími se instancí objektů a tříd.. Hrany v sítích reprezentují různé druhy vztahů mezi uzly – jsou zde vedle sebe ty, které reprezentují *intenzionální znalosti*, spolu s těmi, které reprezentují *extenzionální znalosti* (viz následující odstavec 7.2).

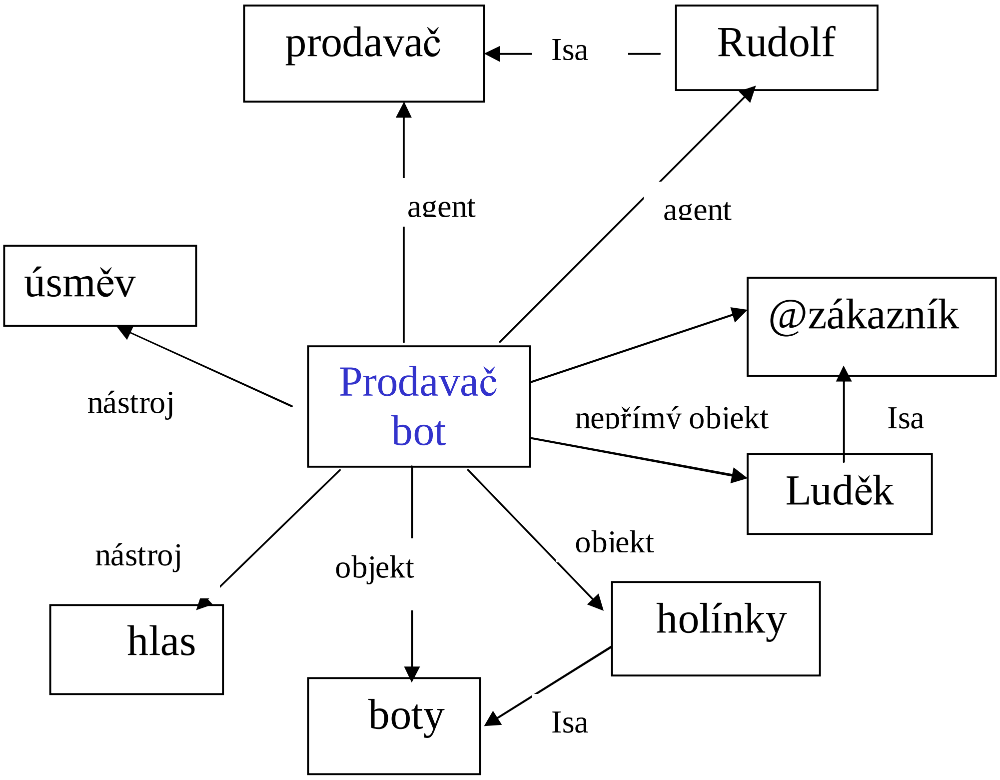
7.1.2 Rámce

Ve druhé polovině minulého století se též rozvíjela myšlenka reprezentace znalostí pomocí rámců (frame), umožňující též lineární zápis obsahu asociativních sítí. Rámec zpravidla reprezentuje koncept (třídu) definovaný identifikátorem a elementy zvanými sloty – každý slot odpovídá atributu, který má mít koncept. Hodnoty atributů jsou buď prvky konkrétní domény nebo identifikátory dalších rámců. Rámec může též reprezentovat samostatné individuum s atributem Isa ve vztahu ke třídě, o jejíž instanci jde. Tyto principy však v rámcích nedisponují dostatečně vyhovujícím odvozovacím systémem. Navíc se rámce, podobně jako asociativní sítě, vyznačují nevyjasněnou sémantikou.

Příklad 7.1

Reprezentace znalostí o prodeji obuvi

a) Reprezentace v asociativní síti



obr. 7.1

b) Reprezentace v rámci

Rámec
Identifikátor rámce : prodavač_bot
Sloty
Agent : prodavač
Agent : Rudolf
Objekt : boty/holínky
Nepřímý objekt : zákazník/Luděk

c) Predikátová reprezentace *A* a její interpretace

∀x (**prodavač_bot**(x) ↔ **prodavač**(x) & ∃y (**zákazník**(y) & ∀z (**prodává**(x, y, z) & **boty**(z))))

Struktura interpretace v predikátové logice:

W = {Rudolf, Eva, Luděk, Milan, Jiří, šaty, holínky… }

D(**zákazník**) = {Eva, Milan, Jiří}

D(**prodává**) = {(Milan, Eva, šaty),….,(Rudolf, Luděk, holínky) }

……

Interpretace formule :

A : **prodává**(Rudolf, Luděk, holínky) & **zákazník**(Luděk)

I(*A*) = ***I***(true & false) = false

a) **Reprezentace (konceptová) v deskripční logice omezená na unární a binární predikáty** (podrobně viz následující lekce) :

prodavač_bot ≡ **prodavač** ⊓ ∃ PRODÁVÁ_KOMU **. zákazník** ⊓ ∀ PRODÁVÁ_CO **. boty**

Struktura interpretace v deskripční logice:

W = {Rudolf, Eva, Luděk, Milan, Jiří, šaty, holínky… }

D(**zákazník**) = {Eva, Milan, Jiří}

D(PRODÁVÁ_KOMU) = {(Milan, Eva),….,(Rudolf, Luděk)}

D(PRODÁVÁ_CO) = {(Milan, šaty),….,(Rudolf, holínky)}

……

7.2 TERMINOLOGICKÁ (KONCEPTOVÁ) REPREZENTACE ZNALOSTÍ

Citát převzatý z úvodu k SUO IFF
(Standard Upper Ontology Information Flow Framework)

Phylosophy cannot become scientifically healthy without an immense technical vocabulary. We can hardly imagine our grand-grandsons turning over the leaves of this dictionary without amusement over the paucity of words with which their grandsires attempted to handle metaphysics and logic. Long before that day, it will have become indispensably requisite, too, that each of this terms should be confined to a single meaning, which, however broad, must be free from all vagueness. This will involve a revolution in terminology; for in its present condition a philosophical thought of any precision can seldom be expressed without lengthy explanations.

Charles Sanders Peirce, Collected Papers 8:169

Společnou myšlenkou koncepce asociativních sítí, rámců a objektového přístupu je reprezentovat znalosti ve formě tříd a vztahů mezi nimi, a to především vztahu množinové inkluze, umožňující zachycení hierarchických struktur. V roce 1977 přišli Woods a Brachman s návrhem nového „konceptového jazyka“ založeného na myšlence, že je třeba rozlišovat obecnou konceptovou úroveň (intenze) od úrovně instancí konceptu (určují extenzi konceptu).

To vedlo k vytvoření systému KL (1985) a řady dalších. Reprezentace znalostí je v těchto systémech založena na *konceptovém* přístupu, který rozlišuje mezi terminologickými znalostmi a tvrzeními. První zacházejí s (koncepty = pojmy = třídami) a jejich vztahy, druhé se týkají individuí a jejich příslušnosti konceptům a jejich vztahům.

7.2.1 Základní pojmy terminologické (konceptové) reprezentace znalostí

V *konceptových jazycích* jsou *koncepty* používány k reprezentaci tříd jako množin prvků a pro účely specifikace jejich vlastností nebo atributů jsou určeny binární relace reprezentující *vztahy* (*role*). Typické koncepty jsou strukturovány do hierarchií.

Definice 7.1

Konceptové paradigma vidí svět jako množinu pojmů – konceptů a jejich vzájemných vztahů
Konceptová reprezentace rozlišuje mezi :

- *terminologickými znalostmi* (Tbox znalostní báze) – týkají se konceptů a jejich vzájemných vztahů
- *tvrzeními o jejich instancích* (Abox znalostní báze) – týkají se příslušnosti objektů konceptům a jejich vztahům

Formální konceptové jazyky disponují predikáty :

- unárními pro reprezentaci *konceptů*
- binárními pro reprezentaci *vztahů* (*rolí*)

Definice 7.2

Koncept (v objektovém pojetí třída) je (event. neukončený) soubor entit

- sdílejících nějaké základní vlastnosti
- vzájemně se lišících v jiných vlastnostech.

7.2.2 Extenzionální a intenzionální sémantika konceptu

Je-li koncept reprezentován unárním predikátem logiky prvního řádu, lze jeho význam definovat

- *extenzí* - výčtem prvků příslušné unární relace, tj. denotátem příslušného predikátu v rámci dané interpretační struktury,
- *intenzí* - formální specifikací jeho vlastností (pomocí formule, resp. množiny formulí) v reprezentujícím jazyce (zde v jazyce predikátové logiky).

Příklad 7.2

Koncept "malé_přirozené_číslo", reprezentovaný unárním predikátem

malé_přirozené_číslo(x)

může mít přiřazen svůj význam jako denotát

D(**malé_přirozené_číslo**(x)) = {1,2,3},

tj. unární relaci v rámci nějaké interpretační struktury. Tento denotát pak určuje *extenzi* predikátu **malé_přirozené_číslo**(x).
Formální specifikací uvedeného konceptu pak může být predikátová formule

$\forall$x (**integer**(x) & 1≤x & x≤3), určující jeho *intenzi*.

7.3 FORMÁLNÍ ONTOLOGIE

7.3.1 Formální ontologie jako oblast informatiky

Formální ontologie navazuje na *tradiční (filosofickou) ontologii* jako teorii bytí.
Formální ontologie se zabývá

- formálními definicemi vlastností, rozdílů a vzájemných vztahů entit (konceptů) v rámci určitého referenčního systému - modelovaného světa
- stanovením univerzálních kategorií (konceptů), které jsou potřebné k tomu, aby se dalo obecně hovořit o tomto světě.

Definice 7.3

Formální ontologie je reprezentace sdílené konceptualizace v určité oblasti, založená na terminologickém slovníku (ontologii) v dané oblasti. (hierarchicky strukturovaných konceptech).

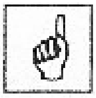
7.3.2 Ontologie jako hierarchická struktura konceptů

Je třeba rozlišit :

- *Specializované ontologie* :
  Týkají se problému stanovení relevantní esencí entit a jejich vzájemných vztahů v rámci konkrétního modelovaného světa - *speciální, aplikační* resp. *doménová ontologie*
- *Obecná ontologie :*
  Řeší otázku, zdali jsou zde entity spolu s jejich vzájemnými vztahy, které jsou svou obecností přítomny v celých třídách modelovaných referenčních systémů – ontologie (nej)vyšší úrovně (*upper* / *top-level ontology*).

Hlavní odlišnosti obecné ontologie od ontologií pro speciální účely :

- Obecná ontologie je víceméně aplikovatelná na každou speciální ontologii (s přidáním axiómů speciálních pro danou oblast).
- Pro různé oblasti jsou zde různé typy reprezentovaných znalostí (měly by být unifikovány, aby bylo možno odvozovat z různých oblastí současně).



3x ONTOLOGIE - pokaždé jiný význam

1. Teorie bytí, která je spolu s teorií poznání tvoří náplň klasické filozofie
2. Oblast v rámci reprezentace znalostí v umělé inteligenci - formalizovaný popis objektů modelovaného světa – referenčního systému
3. Ontologie jako produkt bodu 2. – terminologický slovník hierarchicky uspořádaných pojmů z dané problémové oblasti

7.3.3 Formální logika a formální ontologie

Výše uvedené dva navzájem rozdílné způsoby definování sémantiky konceptu, tj. prostřednictvím jeho extenze nebo jeho intenze, jsou též charakteristické pro dva rozdílné přístupy k reprezentaci znalostí. I když by byl pro reprezentaci konceptu shodně zvolen (event. modifikovaný) jazyk logiky prvního řádu, liší se tradiční logický přístup se svou extenzionální sémantikou predikátů a ostatních jazykových prvků od relativně nového přístupu intenzionálního, známého již z tradiční filozofické ontologie (teorie bytí).

Jedním ze základních rysů sémantiky tradiční formální logiky prvního řádu, je skutečnost, že se zabývá obecnými vlastnostmi různých aplikovatelných (především matematických) struktur interpretace, které mohou být modely tohoto jazyka. Přístup tradiční logiky prvního řádu, a to i tehdy, má-li sloužit jako prostředek reprezentace ve znalostních bázích, je v zpravidla typicky „ontologicky neutrální“, tj. vyznačuje se větším důrazem na formu a formální principy dedukce než na sémantiku ve smyslu ontologicky co nejvěrnější reprezentace znalostí.

Při tradičně logickém přístupu se znalostní báze nějakého referenčního systému vytváří jako formální popis (logickými formulemi) současného stavu tohoto referenčního systému (modelovaného světa). Tím jsou dány i denotáty speciálních jazykových prostředků (predikátů, funkcí, konstant) použitých v této *zamýšlené interpretaci* (obr. 7.2). Soubor logických formulí, reprezentujících současný stav modelovaného světa, tvoří pak soubor *speciálních axiómů* jako základ, z něhož lze podle odvozovacích pravidel v rámci uvažované logiky prvního řádu generovat další *věty teorie*.

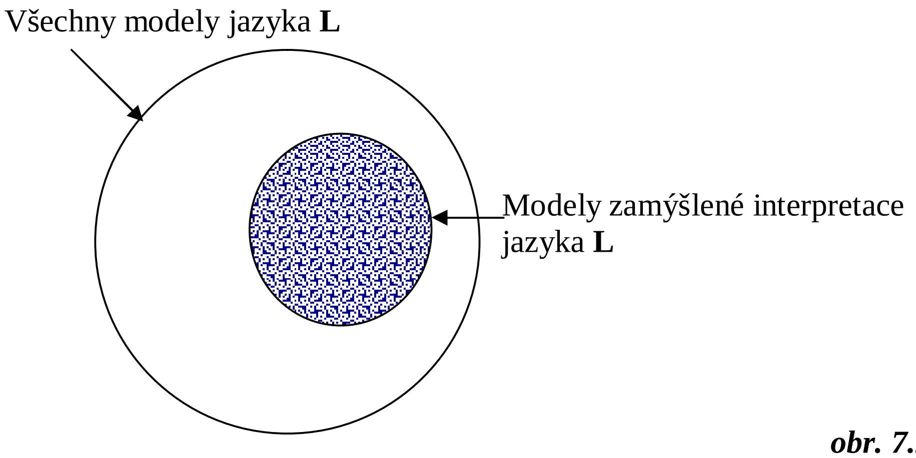
Z hlediska tradiční logiky zamýšlená interpretace není jedinou možnou interpretací vytvořené znalostní báze. Mohou existovat i další modely, tj. struktury interpretací, v nichž platí všechny formule znalostní báze. Mohou tedy existovat i jiné možné světy, pro něž se vytvořená reprezentující znalostní báze hodí. V rámci těchto světů mohou mít i koncepty, reprezentované predikáty ve znalostní bázi, rozdílné extenze.

Ontologický přístup je přístupem formalizované sémantiky, který nezahrnuje tuto relativní interpretační "svévolnost". Při ontologickém přístupu je jazyk logiky prvního řádu zvolen za prostředek reprezentace znalostí tak, aby účelově posloužil určitému konkrétnímu modelování nějakého *referenčního systému*, tj. určitým způsobem vymezeného modelovaného jsoucna. Je tedy třeba, aby jeho modely byly výhradně těmi strukturami, které jsou odvozeny speciálně z vlastností a vztahů entit modelované reality, odpovídajících právě této zamýšlené interpretaci. Starost o obecné vlastnosti ostatních modelů - interpretačních struktur, které nemají nic společného s interpretacemi zamýšlenými, zde tedy zcela odpadá.

> V ohnisku zájmu ontologicky formalizované sémantiky je snaha formálním jazykem co nejlépe a na základě co nejobecnějšího konsensu strukturovaně reprezentovat ty entity a jejich vzájemné vztahy, které představují relevantní esenci zamýšlených modelů daného referenčního systému.

Znalostní báze v případě ontologického přístupu odráží v daném okamžiku pevně stanovený modelovaný svět, neboť jen v případě takové přísné vazby znalostní báze na zamýšlenou interpretaci lze každý koncept popsat stanovenými vlastnostmi, umožňujícími následné stanovení jeho extenze.

Ontologický přístup nevylučuje úvahy o dalších možných světech, ale na rozdíl od přístupu tradičně logického jsou tyto světy svázány s původním světem zamýšlené interpretace relací vzájemných možných přístupů.



obr. 7.2

Na začátku tohoto snažení však evidentně stojí problém stanovení, které entity se svými vzájemnými vztahy jsou pro speciální problém tou relevantní esencí konkrétního modelovaného světa - zde se hovoří o *speciální*, *aplikační* resp. *doménové ontologii* - a zdali jsou zde entity spolu s jejich vzájemnými vztahy, které jsou svou obecností přítomny v celých třídách modelovaných referenčních systémů - v tomto případě jde o *ontologii nejvyšší úrovně* (v literatuře často nazývanou *top-level ontology*).

7.3.4 Sowův návrh ontologie nejvyšší úrovně

Příklad 7.3

Sowův návrh hierarchie kategorií reprezentace znalostí na nejvyšší úrovni vychází z primitivních kategorií

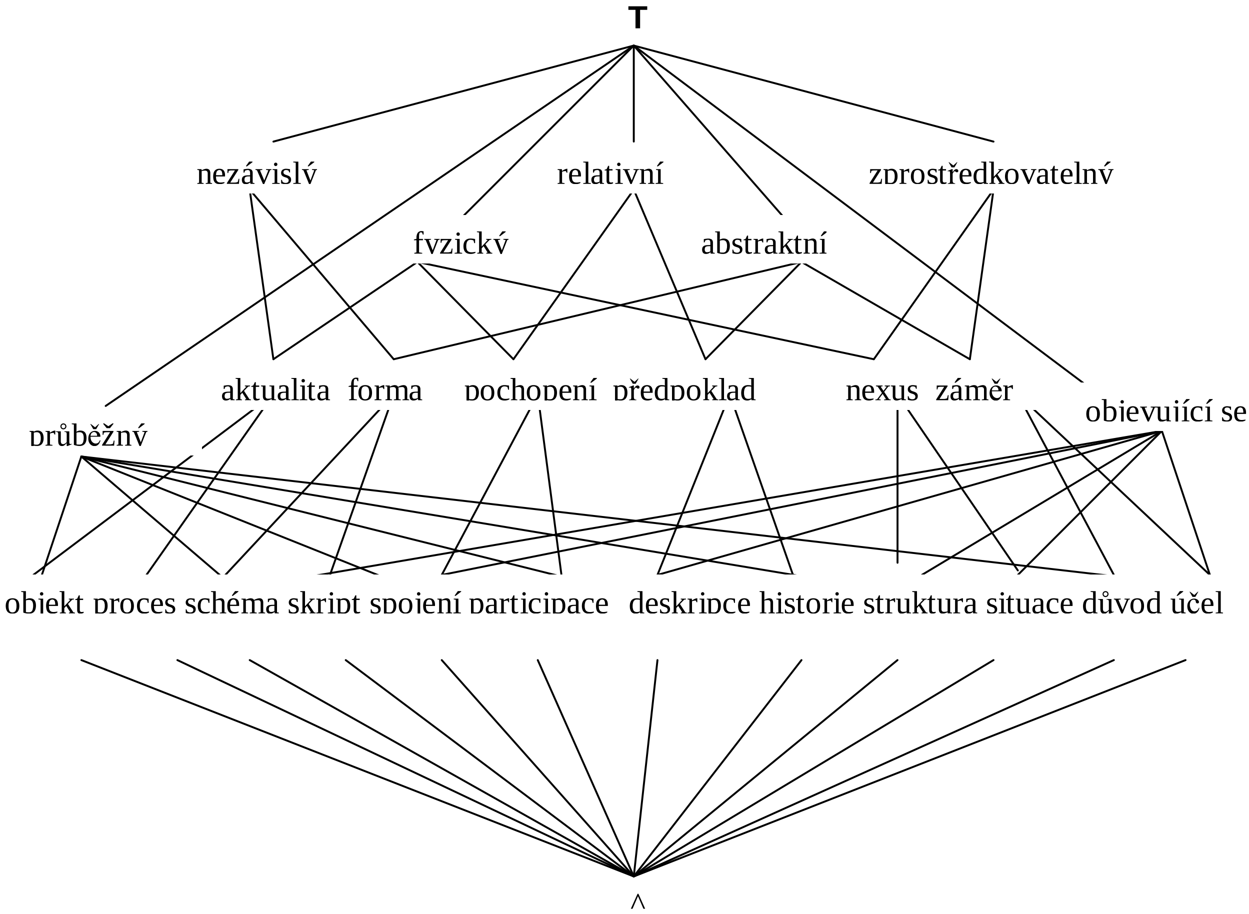
- nezávislý, relativní, zprostředkovaný,
- fyzický, abstraktní
- průběžný, objevující se

V obr. 7.3 je znázorněno 9 základních kategorií – typů konceptů : **T**, ^, nezávislý, relativní, zprostředkovatelný, fyzický, abstraktní), průběžný, objevující se. Každý subtyp je definován pomocí infima (nejvyššího společného subtypu reprezentovaného $\cap$) dvou supertypů, jejichž vlastnosti (axiómy) dědí. Např. koncept forma je nezávislý a abstraktní.

Dvěma základními cestami zobrazení kombinatorických struktur kategorií jsou konceptuální grafy a matice. Následující tabulka zobrazuje matici podle J.W.Sowy.

	fyzický		abstraktní	
	průběžný	objevující se	průběžný	objevující se
nezávislý	objekt	proces	schéma	skript
relativní	spojení	participace	deskripce	historie
zprostředkovaný	struktura	situace	důvod	účel

tab. 7.1



obr. 7.3

7.4 FORMÁLNÍ ONTOLOGIE A IS

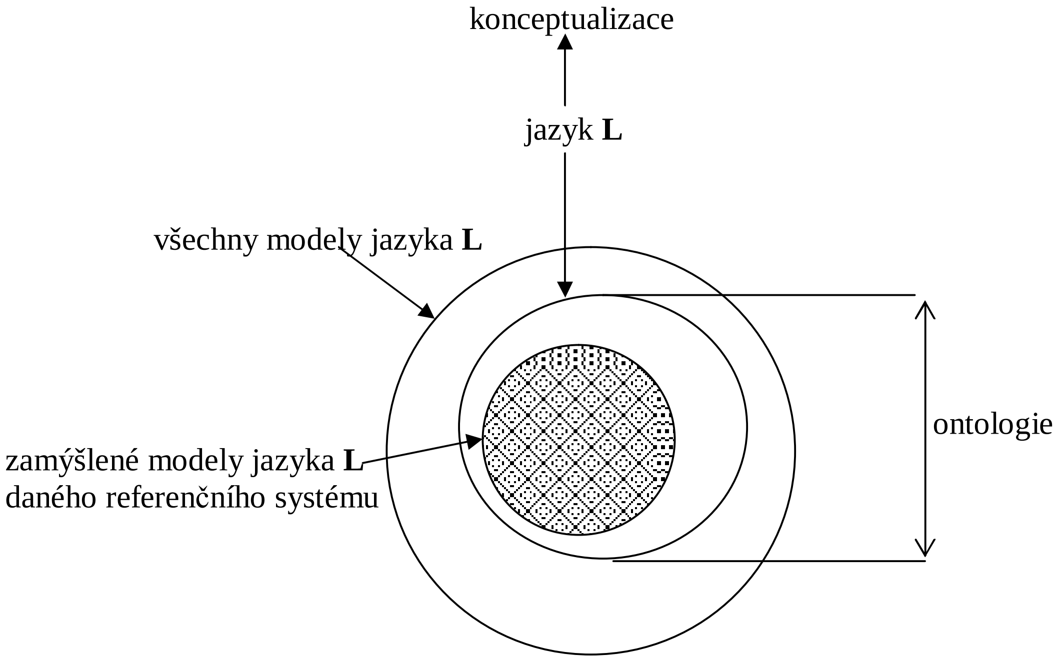
7.4.1 Ontologie a ontologická konceptualizace

Řada odborníků specializovaných na informační a znalostní systémy se shoduje v tom, že největším úspěchem poslední doby v oblasti reprezentace znalostí jsou systémy pro reprezentaci strukturované informace na bázi ontologie - formalizované sémantiky.

V umělé inteligenci znamená ontologie vysoce interdisciplinární přístup, v němž filosofie a lingvistika hrají zásadní roli při analýze struktury formálně reprezentované reality. V takto vymezené roli má formální ontologie perspektivně zásadní význam při vytváření *ontologicky řízených* informačních systémů, vyznačujících se jednotnou metodologií konceptualizace s vysokou mírou všeobecné akceptovatelnosti.

Během procesu konceptualizace, jehož cílem je vytvoření konceptuálního modelu, tvůrce vidí modelovaný svět prostřednictvím určitého paradigmatu. Paradigmatem se přitom rozumí určitý způsob vidění referenčního systému (modelovaného světa). Adekvátně tomuto způsobu vidění tvůrce zvolí abstraktní model, tj. soubor vhodných pojmů a jejich vzájemných vztahů s danou sémantikou. Nástrojem umožňujícím vlastní modelování je pak modelu odpovídající zvolený jazyk. Pojmy a prostředky téhož konceptuálního modelu je možno reprezentovat prostřednictvím celé řady různě zvolených jazykových prostředků. Např. model vycházející z objektového paradigmatu může pro reprezentaci znalostí o modelovaném světě použít stejně dobře symbolických jako grafických jazykových prostředků.

Konceptualizace je formální reprezentace struktury tvořené entitami referenčního systému v jejich vzájemných vztazích, tak jak je vnímána a organizována agentem, prostřednictvím vhodně zvoleného formálního jazyka. Agenti však mohou komunikovat pouze tehdy, jestliže se jejich zamýšlené modely překrývají.



obr. 7.4

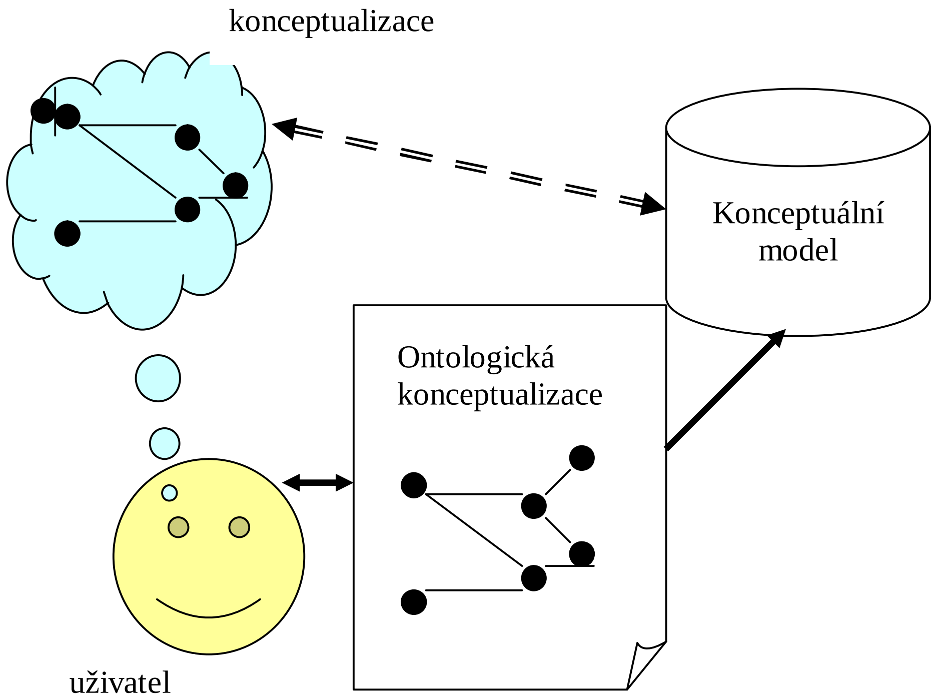
Z potřeby *ontologicky adekvátních znalostních bází* a z nich vycházejících informačních systémů, která je typická pro vývoj posledních let, vyvstala též potřeba nového pojetí konceptuálního modelování – *ontologická konceptualizace*.

Vývoji *ontologicky vedené metodologie konceptuálního modelování* přispěla integrace nových přístupů v databázích, zvláště pak znalostních bázích a OO-systémech, a to speciálně jejich

- Zakotvenost v realitě
- Transparentnost pro uživatele
- Obecnost
- Rigoróznost

Založenost na

- Logice
- Filosofii
- Lingvistice



obr. 7.5

Konceptuální modely vycházející z ontologicky vedené metodologie by měly být přesnější, přirozenější (bližší všeobecně akceptovanému pojetí) a transparentnější, tj. s významem jasným i laickým uživatelům.

7.4.2 Sémantický web

V posledních letech se úsilí specialistů v oblasti formální reprezentace znalostí, inspirované praktickými problémy s informacemi uloženými na webu, soustřeďuje na vytvoření moderních inteligentních systémů schopných na jedné straně obhospodařovat rozsáhlá data a na druhé straně poskytovat mechanismy „inteligentního“ odvozování ze znalostních bází.

Pojem *Sémantický Web* představuje v současné době určitou vizi, jak by měla být významová stránka informací a znalostí na webu reprezentovány prostřednictvím formálních struktur, aby k nim bylo možno přistupovat, a to bez ohledu na rozdílné typy zdrojů (data, služby, webové stránky aj.) prostřednictvím agenta softwarového typu.

Hlavní ideou sémantického webu je docílit možnosti dotazování různého typu na základě speciálního, dostatečně expresivního konceptuálního modelu, schopného popisu vzájemných vztahů mezi základními reprezentačními jednotkami - jmény, symboly. Protože symbol může mít různý význam v rozdílných aplikacích, resp. v různých přirozených jazycích, je nutno v rámci konceptuálního modelu vždy též prostřednictvím definice explicitně vyjádřit jeho význam. Definice významu webových pojmů bývají organizovány do speciálních struktur , kterými jsou ontologie.

V kontextu vize sémantického webu je participace autonomních agentů jednou z jejích základních složek. Přitom vývoj autonomního agenta, který je schopen interakce se softwarovými systémy v různorodých prostředích, je obtížný problém. Je třeba především respektovat tři základní výhledové cíle :

1. Aby bylo možno vyvíjet systémové architektury, které jsou agentům srozumitelné a navíc modifikovatelné a rozšiřovatelné, je třeba je založit na v rámci architektury přesně definovaných pojmech a jejich vzájemných vztazích.
2. Je třeba uvést do souvislostí interně definované pojmy v rámci architektury s pojmy vnějšího hostitelského prostředí, v rámci něhož se mohou odehrávat interakce agenta.
3. Hlavním cílem je implementovat procesy potřebné pro řešení problémů, jakými jsou např. získávání informací, plánování a řízení, konfrontace a vyjednávání apod., to všechno na bázi logických rozhodování. K tomu jsou samozřejmě zapotřebí speciální ontologie.

7.4.3 Konceptuální modelování a ontologie

Při vytváření ontologií pokrývajících vždy určitou doménu je podstatná všeobecná shoda, jak má výsledná ontologie vypadat. Na základě dlouholetých zkušeností se odborníci shodují přinejmenším na tom, že vidění světa jako množiny objektů a jejich vzájemných vztahů je vhodnou bází informačních a znalostních systémů a též mnoha aplikací. Deskripční logika (viz následující lekce) toto paradigma podporuje.

Pro nově vytvářenou ontologii je třeba stanovit množinu jmen bázových konceptů a jejich vztahů - rolí, které v reprezentovaném světě sehrávají. Složitější koncepty jsou pak specifikovány pomocí definic na základě konstruktorů konceptů a rolí.

Hierarchicky strukturovaný terminologický slovník, zvaný též ontologie, představuje konceptuální schéma datového modelu určitého modelovaného světa. Konceptuální datové modely by měly zajistit přesnou a sémanticky bohatou reprezentaci složitých vlastností a vztahů existujících mezi dokumenty. Konceptuální (ontologické) modelování řeší otázku, jak deklarativním a přitom opakovaně použitelným způsobem popsat doménu, do níž jsou situovány vytvářené aplikace, pomocí terminologických slovníků a jak formulovat omezení, umožňující získávat požadovaná data.

V současné době jsou přijímány jako standardy E-R model jako jazyk relačního konceptuálního modelování, UML a ODMG jako jazyky konceptuálního modelování v rámci objektově orientovaných modelů a XML, RDF(S), DAML+OIL a OWL jako jazyky konceptuálního modelování pro webový semi-strukturovaný datový model. Nicméně problém konceptuálního modelování zůstává mezi hlavními problémy v souvislosti s realizací představ o sémantickém webu. Ukazuje se, že deskripční logika (viz následující lekce) se může stát sjednocujícím formalismem výše uvedených přístupů.

7.4.4 Práce se zdroji specifikovanými v RDF a DAML+OIL

Systém OilEd vyvinutý na univerzitě v Manchesteru dává možnost s grafickou podporou vytvářet ontologie a přistupovat k dokumentům v DAML+OIL. S použitím např. systému RACER (univerzita v Hamburgu) lze pak ověřovat konsistenci konceptů hledat modely znalostní báze apod.

Konceptový (terminologický) přístup k reprezentaci znalostí, který vyústil v *konceptové (ontologické) jazyky*, prošel v nedávné minulosti jistými vývojovými fázemi, v nichž podstatnou úlohu sehrály zejména tyto přístupy : asociativní sítě, rámce , predikátová logika, objektová reprezentace.

Principy terminologické reprezentace v konceptových jazycích :
Koncepty jsou v ontologických jazycích používány k reprezentaci terminologických znalostí - tříd (pojmů) jako množin prvků.
Pro účely specifikace vlastností konceptů jejich vzájemných vztahů nebo jejich atributů jsou určeny *role* (*omezení*).
Koncepty jsou uspořádávány do terminologických slovníků - *hierarchických struktur*.
Formální ontologie je reprezentace sdílené konceptualizace v určité oblasti založená na terminologickém slovníku (ontologii) v dané oblasti. (hierarchicky strukturovaných konceptech).
Ontologie jako hierarchická struktura konceptů :
Specializované ontologie řeší problém stanovení relevantní esencí entit a jejich vzájemných vztahů v rámci konkrétního typu modelovaného světa - *speciální, aplikační* resp. *doménová ontologie*.
*Obecná ontologie ř*eší otázku, zdali jsou zde entity spolu s jejich vzájemnými vztahy, které jsou svou obecností přítomny v celých třídách modelovaných referenčních systémů – ontologie (nej)vyšší úrovně (*upper* / *top-level ontology*).
Formální ontologie má perspektivně zásadní význam při vytváření *ontologicky řízených* informačních systémů, vyznačujících se jednotnou metodologií konceptualizace s vysokou mírou všeobecné akceptovatelnosti.

Pojem *Sémantický Web* představuje v současné době určitou vizi, jak by měla být významová stránka informací a znalostí na webu reprezentovány prostřednictvím formálních struktur, aby k nim bylo možno přistupovat, a to bez ohledu na rozdílné typy zdrojů (data, služby, webové stránky aj.) prostřednictvím agenta softwarového typu.

Kontrolní otázky

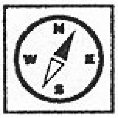
1. Čím je charakteristický konceptový přístup k reprezentaci znalostí ?
2. Co jsou koncepty a jejich role ?
3. Vysvětlete pojem formální ontologie.
4. Co je sémantický web ?

Literatura

1. Klein, M. : Combining and relating ontologies : an analysis of problems and solutions. CEUR-WS, Vol 47
2. Ohlbach, H.J., Koehler, J. : Modal logics, description logics and arithmetic reasoning. Artificial Intelligence 109, 1999, pp 1-31. ¨
3. Pisanelli, D.M., Gangemi, A., Steve, G. Ontologies and Information Systems : The Marriage of the Cerntury. Procc. of Lyee Workshop, Paris, 2002
4. Sanner, S.P. : Towards practical taxonomic classification for description logics on the Semantic Web. http://www-ksl.stanford.edu/pub/ksl_reports/
5. Schaerf, A. : Reasoning with Individuals in Concept Languages. Data and Knowledge Engeneering, Vol. 13(2), pp 141-176.
6. Schmidt-Schau3, M., Smolka, G. : Attributive Concept Descriptions with Complements. Artificial Intelligence Journal 48(1), 1991, pp 1-26.

7. Sowa, J.F. „ Knowledge Representation „ Logical, Philosophical, and Computational Foundations, Brooks Cole Publishing Co., Pacific Growe, CA, 2000, 0-534-94965-7

8 DESKRIPČNÍ LOGIKA I



V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se seznámíte s formálním konceptovým jazykem deskripční logiky, který je odvozen z jazyka logiky prvního řádu pomocí řady zjednodušení. Uvidíte, že jazyk deskripční logiky, opírající se do značné míry o svoji sémantiku, vám bude poměrně blízký a srozumitelný.

8.1 KONCEPTOVÉ JAZYKY A DESKRIPČNÍ LOGIKA

Jak již bylo konstatováno v předcházející lekci, systémy reprezentace znalostí založené na konceptových jazycích se vyznačují tím, že vycházejí z popisu (deskripce) konceptů, uspořádatelných do hierarchických struktur (terminologických znalostí), a jejich vzájemných vztahů. V *konceptových jazycích* jsou *koncepty* používány k reprezentaci tříd jako množin prvků a pro účely specifikace jejich vlastností nebo atributů jsou určeny binární relace reprezentující - *vztahy*.

Deskripční logika (DL) jako logický reprezentant konceptových jazyků bývá charakterizována jako sjednocující formalismus pro reprezentaci v systémech založených na rámcích, asociativních sítích, objektově orientované reprezentaci znalostí, sémantických datových modelech, relačních databázích, konceptových jazycích a v neposlední řadě na snaze o vytvoření ontologického jazyka nejvyšší úrovně.

Z hlediska logiky prvního řádu je DL její strukturovaný fragment, který poskytuje teoretický základ pro reprezentaci strukturované informace ve znalostních bázích, včetně možnosti z nich odvozovat.

Konceptové výrazy jsou v DL vytvářeny pomocí konstruktorů, z nichž některé specifikují vztahy k jiným konceptům (prostřednictvím rolí).

Formální systém deskripční logiky je určen jazykem, znalostní bází, která představuje soubor speciálních axiómů pro dedukci, a pravidly (mechanismem) tuto dedukci umožňujícími.

8.2 JAZYK DESKRIPČNÍ LOGIKY

8.2.1 Syntax jazyka deskripční logiky

Jazyk deskripční logiky je speciální *konceptový jazyk* vygenerovaný množinou konstruktorů pro reprezentaci *pojmů*, tj. *konceptů* (tříd, kategorií), *rolí* (vztahů mezi nimi) a instancí - jejich *individuí*.

Bází konceptového jazyka jsou atomické koncepty – unární predikáty a atomické role – binární predikáty, z nichž se pomocí konstruktorů vytvářejí komponované koncepty a role.

Definice 8.1 (abecedy jazyka deskripční logiky)

Abecedu jazyka deskripční logiky tvoří

- symboly pro jména konceptů - unární predikátové symboly, které reprezentují třídy individuí
- symboly pro jména rolí - binární predikátové symboly, které reprezentují relace mezi páry individuí
- symboly pro jména individuí.

Pro jména individuí platí v DL „*předpoklad jedinečného jména*“, tj. předpoklad, že neexistuje dvojice navzájem různých jmen pojmenovávajících týž objekt.

Gramatická pravidla jsou dána množinou konstruktorů pro vytváření formulí jazyka, tj. definic konceptů a rolí pro vytváření tzv. TBoxů a ABoxů, z nichž sestávají znalostní báze. Gramatiky různých modifikací jazyka deskripční logiky se od sebe liší výběrem konstruktorů. ***FL*** $^-$ **-** jazyky [3] zahrnují konstruktory konjunkce, univerzální kvantifikace a nekvalifikované existenční kvantifikace. ***AL*** $^-$ **-** jazyk [11] je pak rozšířením jazyka ***FL*** $^-$ o konstruktor komplementu a speciální konstruktory $\bot$ a $\top$. Superjazyky jazyka ***FL*** $^-$ (podobně ***AL*** $^-$) jsou vždy určeny řetězcem např. $FL[E][U][C][R][N][O]^-$ (viz následující def.8.2).

V následující definici jsou uvedeny konstruktory, které se zpravidla vyskytují v ***AL-jazycích***, z nichž nejužívanější je ***ALN***.

Definice 8.2 (gramatiky jazyka deskripční logiky Backus–Naurovou formou)

<atomický koncept>::= *A* / *B* /…		
<koncept>	::= <atomický koncept>	
	::= $\top$	
	::= $\bot$	
	::= ¬<koncept>	(*C*)
	::= <koncept> $\sqcap$ <koncept>	
	::= <koncept> $\sqcup$ <koncept>	(*U*)
	::= ∃<role>	
	::= ∃<role>**.**<koncept>	(*E*)
	::= ∀<role>**.**<koncept>	
	::= $\exists^{>n}$ <role>	(*N*)
	::= $\exists^{\leq n}$ <role>	- “ -

<atomická role> ::= *R* / *S* /…		
<role>	::= <atomická role>	
	::= <role>$\sqcap$<role>	(R)
	::= <role> $^-$	(*I*)

<instance konceptu>	::= <koncept>(<individuum>)
<instance role>	::= <role>(<individuum>,<individuum>)
<individuum>	::= a / b /, a_1 /,….

Konceptové formule nepotřebuji proměnné, neboť podle pozice v jednotlivých konstruktech jsou rozlišitelné unární koncepty od binárních rolí.

Konstruktory nemusí být nutně vzájemně nezávislé, jedny mohou simulovat jiné. Využívá se především následujících přepisů :

$\neg(C \sqcup D) \equiv \neg C \sqcap \neg D$

$(C \sqcap D) \equiv \neg C \sqcup \neg D$

$\neg(\forall \boldsymbol{R}.C) \Leftrightarrow \exists\, \boldsymbol{R}.\neg C$

$\neg(\exists \boldsymbol{R}.C) \Leftrightarrow \forall \boldsymbol{R}.\neg C$

8.2.2 Sémantika jazyka deskripční logiky

Definice 8.3

Interpretace jazyka L_D deskripční logiky je určena svou *strukturou* a *interpretačními pravidly*. Přitom strukturu interpretace $I = (\Delta^I, \bullet^I)$ tvoří neprázdná množina Δ^I (universum diskursu) a funkce $\bullet^I$ (denotační funkce interpretace *I*), která zobrazuje každý koncept *C* do podmnožiny $(C)^I \subseteq \Delta^I$ a každou roli *R* do podmnožiny $(R)^I \subseteq \Delta^I \times \Delta^I$, přičemž je $\top^I = \Delta^I$, $\bot^I = \emptyset$. C^I, resp. R^I určuje *extenzi konceptu C*, resp. *extenzi role R*.

Interpretační pravidla (pravidla stanovení *extenzí* jednotlivých typů konstruktů jazyka):

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

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

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

$(\forall R\, .C)^I = \{d_1 \in \Delta^I \mid \forall d_2 : (d_1, d_2) \in R^I \rightarrow d_2 \in C^I\}$

$(\exists R\, .C)^I = \{d_1 \in \Delta^I \mid \exists d_2 : (d_1, d_2) \in R^I \,\&\, d_2 \in C^I\}$

$(\exists^{>n} R)^I = \{d_1 \in \Delta^I \mid \#\{d_2 \mid (d_1, d_2) \in R^I\} > n\}$

$(\exists^{\leq n} R)^I = \{d_1 \in \Delta^I \mid \#\{d_2 \mid (d_1, d_2) \in R^I\} \leq n\}$

$(R^-)^I = \{(d_2, d_1) \mid (d_1, d_2) \in R^I\}$

Poznámky :

Některé varianty jazyka disponují pouze jednodušší podobou výše uvedené definice kvalifikované existenční kvantifikace $(\exists R\, .C)$ a její sémantiky. Konstruktor nekvalifikované existenční kvantifikace má tvar $\exists R$, resp. $\exists R\top$ a význam $(\exists R)^I = \{d_1 \in \Delta^I \mid \exists d_2 : (d_1, d_2) \in R^I\}$, neumožňující omezení výplně slotů.

Charakteristickým rysem DL jsou konstruktory pro stanovení vztahů mezi koncepty a rolemi. Konstruktory $\forall \boldsymbol{R}.C$, $\exists \boldsymbol{R}.C$ provází hodnotové omezení vyjadřující typy objektů, které mohou vyplňovat role (pro rozlišení funkcí atributů v relacích individuální objekt tvořící druhý atribut je zpravidla nazýván výplní role). Dalším typickým omezením je omezení (*N*) kardinality množin výplní rolí.

8.2.3 Subsumpce konceptů

Definice 8.4

Nechť *C* a *D* jsou koncepty. *C* je *subsumován D*, $C \sqsubseteq D$, je-li $C^I \subseteq D^I$ pro každou interpretaci ***I*** , *C* je *ekvivalentní D*, jestliže $C^I \equiv D^I$ pro každou interpretaci ***I***.

Příklad 8.1

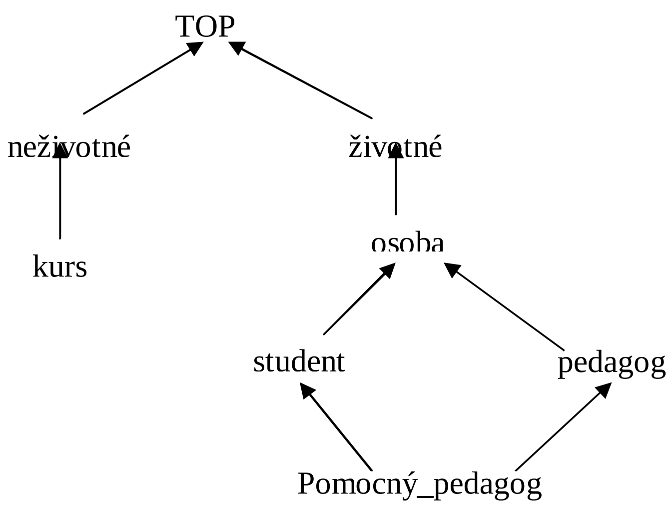
Příklady subsumpcí konceptů :

1. dospělý $\sqcap$ muž $\sqsubseteq$ dospělý
2. dospělý $\sqcap$ bohatý $\sqcap$ muž $\sqsubseteq$ dospělý $\sqcap$ muž
3. $\forall$ DÍTĚ **.** (dospělý $\sqcap$ muž) $\sqsubseteq$ $\forall$ DÍTĚ **.** dospělý
4. $\exists$ DÍTĚ $\not\sqsubseteq$ $\forall$ DÍTĚ **.** dospělý

Subsumpce představuje v prostoru konceptů relaci částečné uspořádání. Základem hierarchické klasifikace je pak pojem taxonomie, představující minimální subsumpci, která je reflexivně-tranzitivním uzávěrem v prostoru konceptů. Klasifikace je zařazení konceptu do taxonomie.

Příklad 8.2

Příklad taxonomie



obr. 8.1

8.3 ZNALOSTNÍ BÁZE V DESKRIPČNÍ LOGICE

***Definice 8.5** (znalostní báze)*

Je-li dán konceptový jazyk **L**, je *znalostní bází* Σ dvojice $\Sigma = (\mathbf{T}, \mathbf{A})$, kde **T** je TBox a **A** je ABox v **L**.

8.3.1 TBoxy

Definice 8.6 *(TBoxu)*

TBox je *terminologický box*, který představuje množinu (významem univerzálně kvantifikovaných) formulí jazyka **L**, jimiž se reprezentují *intenzionální znalosti*. Platí, že

- Formule (termíny) v TBoxu jsou výhradně definicemi, resp. specifikacemi některého z typů uvedených v tab.1,
- v TBoxu se objeví nejvýše jedna definice (specifikace) konceptu určitého jména,
- definice jsou (v základních typech DL jazyků) acyklické.

Přehled syntaxe a sémantiky typů formulí TBoxů (*A*, *C*, *D* jsou koncepty)

	Typ tvrzení TBoxu	**syntax**	**sémantika**
(1)	specifikace primitivního konceptu	$A \sqsubseteq C$	$A^I \subseteq C^I$
(2)	definice primitivního konceptu	$A \equiv C$	$A^I = C^I$
(3)	subsumpce konceptů	$C \sqsubseteq D$	$C^I \subseteq D^I$
(4)	rovnost konceptů	$C \equiv D$	$C^I = D^I$

tab. 8.1

Podle (2) se definují primitivní koncepty TBoxů. Primitivní TBoxy jsou tvořeny pouze těmito typy konstruktorů. *Konceptové jméno* (*termín*) *A* se vždy objeví na levé straně definice, *bázový symbol* (koncept) na pravé straně. *A* zde ***užívá*** *C* ke své definici. V případě znalostní báze s cykly *A* užívá sama sebe (zde se další úvahy omezují na acyklické znalostní báze). *Bázové* (*atomické*) koncepty se nevyskytují na levých stranách formulí.

Příklad cyklické definice termínu : rodič $\doteq$ osoba $\sqcap$ $\exists$ MÁ_DÍTĚ**.**osoba

Definice 8.7 (modelu a splnitelnosti konceptu, resp. role)

Struktura interpretace ***I*** je *modelem* konceptu *C*, resp. role *R*, je-li C^I, resp. R^I neprázdná. Koncept (role) je *splnitelný*(*-á*), má-li model, jinak je *nesplnitelný*(*-á*).

Definice 8.8

Struktura interpretace ***I*** je modelem TBoxu **T**, jestliže splňuje všechny formule z **T**. **T** je splnitelná, má-li model.

Definice 8.9

Bázová interpretace TBoxu je taková, která interpretuje pouze bázové koncepty.

8.3.2 ABoxy

Zatímco formule TBoxu koncepty a jejich role definují v jejich vzájemných vztazích, formule ABoxu vyjadřují fakta neboli tvrzení o jednotlivých konkrétních instancích definovaných konceptů nebo rolí a reprezentují tak extenzionální složku znalostní báze.

Definice 8.10 *(ABoxu)*

ABox (assertional box) je box formulí reprezentujících *extenzionální znalosti*. Formule ABoxu má některou z forem uvedených v tab.2, kde C je koncept, R je role a a, b označují individua jazyka **L**.

Přehled syntaxe a sémantiky typů formulí ABoxů (C je koncept, R je role)

Typ tvrzení TBoxu	**syntax**	**sémantika**
příslušnost ke konceptu	C(a)	$a^I \in C^I$
příslušnost k roli	R(a,b)	$(a^I, b^I) \in R^I$

tab. 2

Konstrukce extenzionální komponenty znalostní báze, tj. ABoxu se realizuje použitím zavedených konceptů a rolí z TBoxu ve tvrzeních o individuích. Jinak řečeno, ABox lze vytvářet až tehdy, jsou-li již zavedeny příslušné koncepty a role v TBoxu a je-li též prověřena konsistence TBoxu.

Příklad 8.3

Tbox :

žena ≡ osoba ⊓ ženského_rodu

muž ≡ osoba ⊓ ¬žena

matka ≡ žena ⊓ ∃ MÁ_DÍTĚ**.**osoba

otec ≡ muž ⊓ ∃ MÁ_DÍTĚ**.**osoba

rodič ≡ matka ⊔ otec

babička ≡ matka ⊓ ∃ MÁ_DÍTĚ**.**rodič

matka_více_dětí ≡ osoba ⊓ $\exists^{\geq 3}$ MÁ_DÍTĚ

manželka ≡ žena ⊓ ∃ MÁ_LEGAL_PARTNERA**.**muž

matka_bez_dcery ≡ matka ⊓ ∀ MÁ_DÍTĚ **.**¬ ženský_rod

ABox :

matka(Marie)

matka_více_dětí(Marie)

otec(Petr)

má_dítě(Marie, Petr)

má_dítě(Marie, Pavel)

má_dítě(Petr, Libor)

Formule ABoxu připomínají jednotlivé instance (řádky) relační databáze s pouze unárními a binárními relacemi. Je zde však podstatný rozdíl, pokud jde o sémantiku. Zatímco instance relační databáze je konsistentní s jejím modelem, formule ABoxu, které jsou bázovými instancemi určitého konceptu, mohou být konsistentní s různými modely tohoto konceptu (téhož schématu databáze). Z chybějících prvků modelu konceptu nelze odvodit negativní tvrzení, neboť zde neplatí tzv. *předpoklad uzavřeného světa*, naopak se zde hovoří o sémantice *otevřeného světa*, kdy absence bázové instance konceptu pouze znamená absenci příslušné informace.

Pokud jde o sémantiku formulí ABoxu, je vždy odkázána, jak je vidět i z tab. 8.2, na sémantiku příslušného konceptu nebo role, jejichž je instancí.

Definice 8.11

Struktura interpretace ***I*** je *modelem ABoxu* **A**, je-li v ní splněno každé tvrzení z **A**. **A** je *konsistentní*, má-li model.

ABox **A** je *konsistentní vzhledem k TBoxu* **T**, existuje-li struktura interpretace, která je současně modelem **A** i **T**.

Je zřejmé, že ABox **A** je konsistentní, je-li konsistentní vzhledem k prázdnému TBoxu.

8.3.3 Konsistence znalostní báze

Definice 8.12

Struktura interpretace *I* je *modelem znalostní báze* Σ = (**T**, **A)**, jestliže je v ní splněna každá formule z **T** i **A**. Znalostní báze Σ je *splnitelná* (*konsistentní*), má-li model.

Příklad 8.4

a) Reprezentace následujících tvrzení

„Student Jan Nový, adresa Zahradní 5, Ostrava 1, pracuje jako pomocný pedagog. Studentem je přitom každý, kdo má zapsán nějaký kurs, pedagogem je každý, kdo vyučuje nějaký kurs.“
"Vyučujícím asistentem může být student postgraduálního studia nebo pedagog."
v predikátové logice a DL :

<u>Predikátová reprezentace</u>

$\forall$x (student(x) $\rightarrow$ (osoba(x) & $\exists$y jméno(x, y) & string(y) & $\exists$z adresa(z) & string(z) & $\exists$w zapsán(x, w) & kurs(w)))

$\forall$x (student(x) $\rightarrow$ $\exists$y zapsán(x,y) & kurs(y))
$\forall$x (pedagog(x) $\rightarrow$ $\exists$y vyučuje(x,y) & kurs(y))

$\forall$x (pomocný_pedagog(x) $\rightarrow$ student(x) & pedagog(x))

student(„Jan Nový“), zapsán(„Jan Nový“, DL02), kurs(DL02)

<u>Reprezentace v DL</u>

student $\equiv$ osoba $\sqcap$ $\exists$ JMÉNO **.** string $\sqcap$ $\exists$ ADRESA **.** string $\sqcap$ $\exists$ ZAPSÁN **.** kurs

pomocný_pedagog $\doteq$ student $\sqcap$ pedagog
jsou formule typu (2) definující koncepty student a pomocný pedagog.

student $\sqsubseteq \exists$ ZAPSÁN . kurs pedagog $\sqsubseteq \exists$ VYUČUJE . kurs

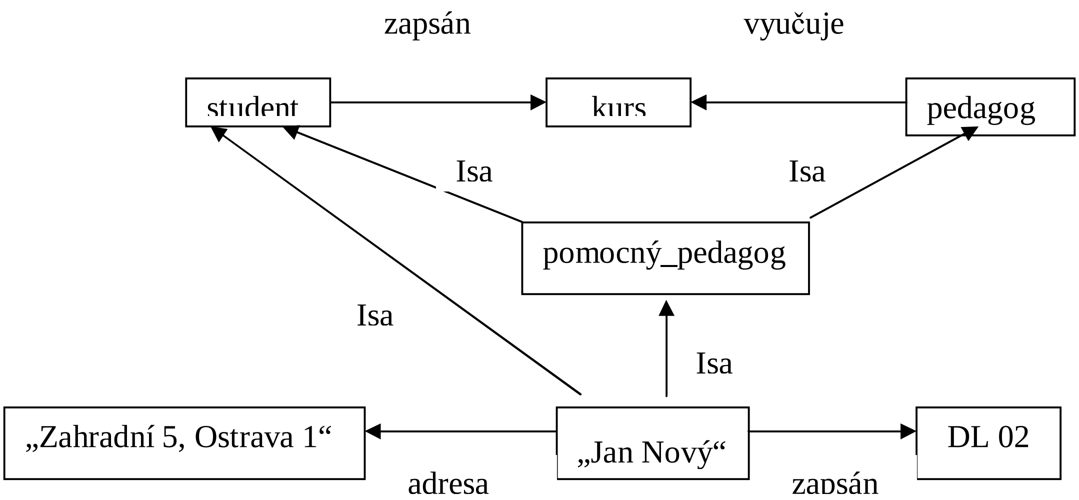
pomocný_pedagog $\sqsubseteq$ student pomocný_pedagog $\sqsubseteq$ pedagog

vyučující_asistent $\sqsubseteq$ postgrad_student $\sqcup$ pedagog
jsou formule typu (3) – inkluze konceptů.

student(„Jan Nový“), ZAPSÁN(„Jan Nový“, DL02), kurs(DL02) jsou formulemi ABoxu – instance konceptů student a kurs a role ZAPSÁN.

Příklad 8.5

Reprezentace znalostí z příkladu (8.4) asociativní sítí



obr. 8.2

Jak je vidět z obr. 8.2, reprezentace asociativní sítí neodděluje extenzionální a intenzionální znalosti.

Úkol 8.1

Vyjádřete extenze následujících komponovaných konceptů na základě extenzí vyskytujících se atomických konceptů :

1. dospělý $\sqcap$ muž
2. dospělý $\sqcap$ bohatý $\sqcap$ muž
3. $\forall$ DÍTĚ **.** (dospělý $\sqcap$ muž)
4. $\exists$ DÍTĚ
5. $\exists$ DÍTĚ $\sqcap$ $\forall$ DÍTĚ **.** ($\exists$ DÍTĚ $\sqcap$ dospělý)

Formální systém deskripční logiky je určen jazykem, znalostní bází, která představuje soubor speciálních axiómů pro dedukci, a pravidly (mechanismem) tuto dedukci umožňujícími.
Jazyk deskripční logiky je speciální *konceptový jazyk* vygenerovaný množinou konstruktorů

pro reprezentaci *pojmů*, tj. *konceptů* (tříd, kategorií), *rolí* (vztahů mezi nimi) a instancí - jejich *individuí*.
Abecedu jazyka deskripční logiky tvoří

- symboly pro jména konceptů - unární predikátové symboly, které reprezentují třídy individuí
- symboly pro jména rolí - binární predikátové symboly, které reprezentují relace mezi páry individuí
- symboly pro jména individuí.

Pro jména individuí platí v DL *„předpoklad jedinečného jména“*, tj. předpoklad, že neexistuje dvojice navzájem různých jmen pojmenovávajících týž objekt.

Gramatická pravidla jsou dána množinou konstruktorů pro vytváření formulí jazyka, tj. definic konceptů a rolí pro vytváření tzv. TBoxů a ABoxů, z nichž sestávají znalostní báze.
Je-li dán konceptový jazyk **L**, je *znalostní bází* Σ dvojice Σ = (**T**, **A**), kde **T** je TBox a **A** je ABox v **L**.

TBox je *terminologický box*, který představuje množinu (významem univerzálně kvantifikovaných) formulí jazyka **L**, jimiž se reprezentují *intenzionální znalosti*.

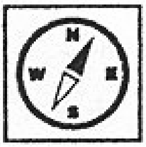
ABox (assertional box) je box formulí reprezentujících *extenzionální znalosti*. Formule ABoxu má některou z forem uvedených v tab.2, kde *C* je koncept, *R* je role a a, b označují individua jazyka **L**.
Interpretace jazyka L_D deskripční logiky je určena svou *strukturou* a *interpretačními pravidly*. Přitom strukturu interpretace $I = (\Delta^I, \bullet^I)$ tvoří neprázdná množina Δ^I (universum diskursu) a funkce $\bullet^I$ (denotační funkce interpretace *I*), která zobrazuje každý koncept *C* do podmnožiny $(C)^I \subseteq \Delta^I$ a každou roli *R* do podmnožiny $(R)^I \subseteq \Delta^I \times \Delta^I$, přičemž je $\top^I = \Delta^I$, $\perp^I = \emptyset$. C^I, resp. R^I určuje *extenzi konceptu C*, resp. *extenzi role R*.
Interpretační pravidla (pravidla stanovení *extenzí* jednotlivých typů konstruktů jazyka):

$$(\neg C)^I = \Delta^I \setminus C^I$$
$$(C \sqcap D)^I = C^I \cap D^I$$
$$(C \sqcup D)^I = C^I \cup D^I$$
$$(\forall R\,.C)^I = \{d_1 \in \Delta^I \mid \forall d_2 : (d_1, d_2) \in R^I \rightarrow d_2 \in C^I\}$$
$$(\exists R\,.C)^I = \{d_1 \in \Delta^I \mid \exists d_2 : (d_1, d_2) \in R^I \,\&\, d_2 \in C^I\}$$
$$(\exists^{>n} R)^I = \{d_1 \in \Delta^I \mid \#\{d_2 \mid (d_1, d_2) \in R^I\} > n\}$$
$$(\exists^{\leq n} R)^I = \{d_1 \in \Delta^I \mid \#\{d_2 \mid (d_1, d_2) \in R^I\} \leq n\}$$
$$(R^-)^I = \{(d_2, d_1) \mid (d_1, d_2) \in R^I\}$$

Struktura interpretace ***I*** je *modelem* konceptu *C*, resp. role *R*, je-li C^I, resp. R^I neprázdná. Koncept (role) je *splnitelný*(*-á*), má-li model, jinak je *nesplnitelný*(*-á*).
Struktura interpretace ***I*** je modelem TBoxu **T**, jestliže splňuje všechny formule z **T**. **T** je splnitelná, má-li model.

Kontrolní otázka :
Jaké jsou hlavní rozdíly v syntaxi jazyků predikátové a deskripční logiky?

9 DESKRIPČNÍ LOGIKA II



Tato lekce, jejíž prostudování by vám mělo trvat zhruba 1,5 h, navazuje na lekci předcházející s tím, že se věnuje formálním postupům odvozování a dokazování, které jsou podstatné pro tvorbu znalostních systémů, založených na deskripční logice.

9.1 ROZHODOVÁNÍ / DOKAZOVÁNÍ V DL

Deskripční logika je axiómatickým systémem. To znamená, že rozhodování, resp. dokazování znalostí v DL systému je deduktivní proces vycházející z obou částí znalostní báze $\Sigma = (\mathbf{T}, \mathbf{A})$, která představuje pro tyto postupy výchozí množinu speciálních axiómů, a že lze pomocí formálních pravidel odvozovat ze znalostní báze další implicitně skryté znalosti. Např. ze znalostní báze v příkladě 8.3 lze odvodit, že Marie je babička.

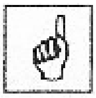
9.2 KONCEPTOVÁ SPLNITELNOST A LOGICKÝ DŮSLEDEK

Definice 9.1

Rozhodování *konceptové splnitelnosti*, tj. dokazování $\Sigma \not\models C \equiv \bot$ je problémem zjišťování existence modelu $\boldsymbol{I}$ znalostní báze Σ takového, v němž $C^{I} \neq \emptyset$.

V jazycích s konstruktorem negace konceptu lze též problém *subsumpce* převést na konceptovou splnitelnost, neboť platí, že $\Sigma \models C \sqsubseteq D$, právě když $\Sigma \models C \sqcap \neg D \equiv \bot$.

Podobně jako v předchozím případě jde i zde o zachování konsistence znalostní báze. Je k tomu potřeba prověřovat u každého rozšíření ABoxu o další instanci, zdali v některém modelu znalostní báze nevede ke sporu.



Při rozšiřování konsistentní znalostní báze o další formule TBoxu je třeba zajistit, aby i po rozšíření zůstala znalostní báze konsistentní.

Definice 9.2

Kontrolou instancí ($\Sigma \models C(\mathrm{a})$) se rozumí zjišťování, zdali tvrzení $C(\mathrm{a})$ je splněno v každém modelu Σ.

V jazycích vybavených konstruktorem negace konceptu (komplementu) lze kontrolu instancí převést na problém konsistence znalostní báze, neboť platí $\Sigma \models C(\mathrm{a})$, kde a je individuum, právě když $\Sigma \cup \{\neg C(\mathrm{a})\} \models \bot$.

9.3 ZNALOSTNÍ BÁZE V DL

Definice 9.3 (logického důsledku znalostní báze)

Formule φ je logickým důsledkem znalostní báze Σ, tj. $\Sigma \models \varphi$, je-li splněna ve všech modelech Σ.

V jazycích vybavených konstruktorem negace konceptu (komplementu) důkaz logického důsledku převést na problém konsistence znalostní báze rozšířené o negaci dokazované formule, neboť platí $\Sigma \models \varphi$, právě když $\Sigma \cup \{\neg\varphi\} \models \bot$. Kontrola instancí je speciálním případem důkazu logického důsledku.

Základem odvozování ze znalostní báze Σ je zjišťování, zdali vztah (např. subsumpce) mezi dvěma konceptovými výrazy je logickým důsledkem jejich deklarací v Σ.

9.3.1 Tablové formální dokazování ze znalostní báze

Postup důkazu logického důsledku, založený na sémantickém tablu, spočívá v důkazu nesplnitelnosti znalostní báze rozšířené o negaci dokazované formule.

Tablový důkaz splnitelnosti konceptové formule nebo množiny konceptových formulí je formálním důkazem, který spočívá v aplikaci tablových pravidel, generujících jejich modely. Nemožnost nalezení modelu svědčí o nesplnitelnosti této množiny formulí.

Tablová odvozovací pravidla jsou odvozena ze sémantiky konstruktorů a jsou generického charakteru, tj. generují postupně formule charakterizující model, pokud nevygenerují spor (nesplnitelný komplementární pár formulí). Např. ze sémantiky konstruktoru $\exists R\,.C$ lze odvodit, že existuje nějaký prvek a universa diskursu takový, že platí $a \in (\exists R\,.C)^I$. Ze sémantiky tohoto konstruktoru dále plyne, že existuje prvek b takový, že $(a, b) \in R^I$ a $b \in C^I$. To vede k následujícímu zobecněnému odvozovacímu pravidlu pro konstrukty kvalifikované existenční kvantifikace (**S** je zde struktura postupně generovaného modelu, jazyk DL je pro účel tablového důkazu modifikován proměnnými v závorkách za příslušnými symboly konceptů a rolí) :

$\mathbf{S} \longrightarrow_{\exists} \{ R(x,y), C(y)\} \cup \mathbf{S}$, jestliže 1. $\exists\, R\,.\, C(x)$ je v **S**, 2. y je nová proměnná,
3. neexistuje z takové, že $R(x,z)$ a $C(z)$ jsou v **S**.

Podobně jsou definována pravidla pro konstruktory univerzální kvantifikace, konjunkce a disjunkce :

$\mathbf{S} \longrightarrow_{\forall} \{C(y)\} \cup \mathbf{S}$, jestliže 1. $\forall\, R\,.\, C(x)$ je v **S**, 2. $R(x,y)$ je v **S**, 3. $C(y)$ není v **S**.

$\mathbf{S} \longrightarrow_{\sqcap} \{C(x), D(x)\} \cup \mathbf{S}$, jestliže 1. $(C \sqcap D)(x)$ je v **S**, 2. $C(x)$ a $D(x)$ nejsou současně v **S**.

$\mathbf{S} \longrightarrow_{\sqcup} \{E(x)\} \cup \mathbf{S}$, jestliže 1. $(C \sqcup D)(x)$ je v **S**, 2. ani $C(x)$ a $D(x)$ nejsou v **S**. 3. $E = C$ nebo $E = D$.

Příklad 9.1

Tablový důkaz nesplnitelnosti formule : student $\sqcap\ \forall$ ZAPSÁN **.** kurs $\sqcap\ \exists$ ZAPSÁN **.** $\neg$kurs

$(\text{student} \sqcap \forall \text{ZAPSÁN}.\text{kurs} \sqcap \exists \text{ZAPSÁN}.\neg\text{kurs})\ (x)$

$\text{student}(x),\ \forall \text{ZAPSÁN}.\text{kurs}(x),\ \exists \text{ZAPSÁN}.\neg\text{kurs}(x) \qquad \sqcap$

$\text{student}(x),\ \forall \text{ZAPSÁN}.\text{kurs}(x),\ \exists \text{ZAPSÁN}.\neg\text{kurs}(x),\ \text{ZAPSÁN}(x,y),\ \neg\text{kurs}(y) \qquad \exists$

$\text{student}(x),\ \forall \text{ZAPSÁN}.\text{kurs}(x),\ \text{kurs}(y),\ \exists \text{ZAPSÁN}.\neg\text{kurs}(x),\ \text{ZAPSÁN}(x,y),\ \neg\text{kurs}(y) \quad \forall$

$\times$

obr. 9.1

Příklad 9.2

Tablový důkaz subsumpce v jazyce s negací s využitím vztahu
$C \sqsubseteq D$, právě když $C \sqcap \neg D$ je nesplnitelný :

Důkaz $(\exists R.A) \sqcap (\exists R.B) \sqsubseteq \exists R.(A \sqcap B)$ pomocí nesplnitelnosti

$(\exists R.A) \sqcap (\exists R.B) \sqcap \neg\exists R.(A \sqcap B)$, tj.

$\exists R.A(x),\ \exists R.B(x),\ \forall R.(\neg A \sqcup \neg B)(x) \qquad 3\text{x} \sqcap$

$\exists R.A(x),\ R(x,c),\ A(c),\ \exists R.B(x),\ R(x,d),\ B(d),\ \forall R.(\neg A \sqcup \neg B)(x) \quad 2\text{x} \exists$

$\exists R.A(x), R(x,c), A(c), \exists R.B(x), R(x,d), B(d),\ \forall R.\neg A(x) \sqcup \forall R.\neg B(x) \quad \forall$

$\exists R.A(x), R(x,c), A(c),$
$\exists R.B(x), R(x,d), B(d),$
$\forall R.\neg A(x)$

$\exists R.A(x), R(x,c), A(c),$
$\exists R.B(x), R(x,d), B(d),$
$\forall R.\neg A(x),\ \neg A(c),$

$\times$

$\exists R.A(x), R(x,c), A(c), \quad \sqcup$
$\exists R.B(x), R(x,d), B(d),$
$\forall R.\neg B(x)$

$\exists R.A(x), R(x,c), A(c), \quad \forall$
$\exists R.B(x), R(x,d), B(d),$
$\forall R.\neg B(x)\ \neg B(d)$

$\times$

obr. 9.2

9.3.2 Dotazování na znalostní bázi

Příklad 9.3

Agent má za úkol najít lidi, kteří by mohli být potenciálními zájemci o koupi určitého typu auta.

TBox: muž $\sqsubseteq$ člověk

žena $\sqsubseteq$ člověk

člověk $\sqcap$ ($\exists$ MÁ_DÍTĚ . člověk) $\sqsubseteq$ rodič

….

ABox: karel : muž

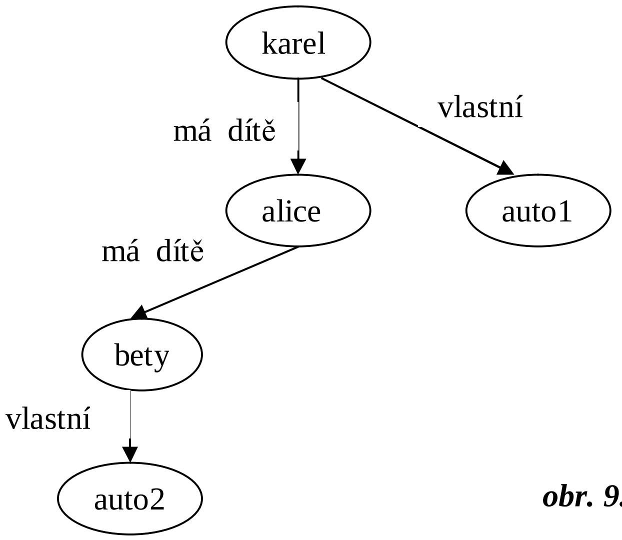
(karel, alice) : MÁ_DÍTĚ

(karel, auto1) : VLASTNÍ

(bety, auto2) : VLASTNÍ

auto1 $\sqsubseteq$ mercedes

auto2 $\sqsubseteq$ bmv



obr. 9.3

Dotaz na instanci ABoxu by mohl vypadat takto :

člověk $\sqcap$ ($\exists$ MÁ_DÍTĚ . rodič) $\sqcap$ ($\exists$ VLASTNÍ . (bmv $\sqcup$ mercedes))

V případě této znalostní báze bude odpovědí "karel".

Dalším dotazem by mohlo být, zdali existuje instance role MÁ_DÍTĚ taková, že

(karel, ($\exists$ MÁ_DÍTĚ . ($\exists$ MÁ_DÍTĚ . žena) $\sqcup$ ($\exists$ MÁ_POTOMKA .($\neg$žena $\sqcap$ ($\exists$ VLASTNÍ . bmv)))) : VLASTNÍ

Odpovědí je ano a je to bety.

9.3.3 Strukturální srovnávání - odvozování ze syntaxe

V DL jazycích, které nedisponují konstruktorem negace, nelze problém subsumpce redukovat na problém splnitelnosti, tak jak to bylo nastíněno v souvislostech s definicí 9.1 rozhodování konceptové splnitelnosti.

Strukturální odvozování je založeno na myšlence, že pokud dvě srovnávané konceptové formule sestávají z podformulí, lze vzájemně srovnávat tyto podformule. Algoritmus má dvě fáze :

1. Normalizace :

   - zploštění uzávorkovaných konjunkcí, tj. přepis $A \sqcap (B \sqcap C)$ na $A \sqcap B \sqcap C$,

   - faktorizace tj. přepis $\forall P.C \sqcap \forall P.D$ na $\forall P.(C \sqcap D)$.

2. Nechť $C = C_1 \sqcap C_2 \sqcap \ldots\ldots \sqcap C_m$, $D = D_1 \sqcap D_2 \sqcap \ldots\ldots \sqcap D_n$ v normální formě. Potom D subsumuje C, právě když platí pro všechna D_i

   a) je-li D_i atomický koncept nebo koncept tvaru $\exists P$, potom existuje C_j takové, že $D_i = C_j$

   b) je-li D_i koncept tvaru $\forall P.D'$, potom existuje C_j tvaru $\forall P.C'$, (atomická role P) takové, že D' subsumuje C'.

Matching:

Jde o vyhledávání shody konceptů se vzory založené na konceptově orientované (concept-centred) normální formě spíše než na standardní subsumpci konceptů, v jazyce *ALN*.

9.3.4 Odvozování z ABoxu

Aby bylo možno rozhodovat konsistenci množiny formulí ABoxu **A**, je třeba vycházet z extenzí konceptů a rolí, definovaných v TBoxu **T**, jejichž instance tvoří **A**. K tomu se využívá tzv. *expandovaný ABox* **A'**, vytvořený ze všech možných instancí dotyčných konceptů a rolí přes všechny prvky jejich extenzí. Kontrolu konsistence **A** vzhledem k **T**, tak jak byla definována (def. 9.1), lze pak převést na kontrolu konsistence samotné **A'**, resp. kontrolu konsistence **A'** vzhledem k prázdnému TBoxu.

Např. nekonsistence způsobená současným výskytem instancí otec(jana) a matka(jana) v ABoxu by nebyla odhalena, kdyby se neuvažovaly extenze konceptů otec a matka.

Problém kontroly instancí ABoxu, tak jak byl definován v odstavci 9.2 se řeší pomocí A'.

9.3.5 Rozšíření znalostní báze podle spouštěcích pravidel

Některé systémy DL zahrnují též tzv. *spouštěcí pravidla*, obsahující pravidla obecného tvaru $C \Rightarrow D$, pomocí nichž lze znalostní bázi rozšiřovat o další části ABoxu podle tohoto procedurálního pravidla :

Platí-li $\Sigma \models C(\mathrm{a})$, přidej do ABoxu i $D(\mathrm{a})$ pro všechny denotáty konstanty <u>a</u> extenze konceptu C.

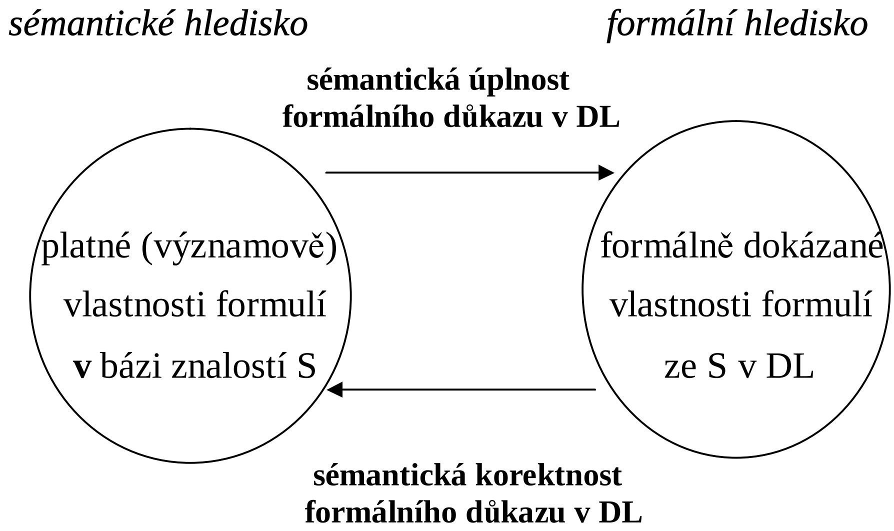
Procedurální rozšíření znalostní báze je možno definovat též deklarativním způsobem. K tomu je třeba zavést epistemický operátor $\boldsymbol{K}$ obohacující TBox znalostní báze o definici epistemického konceptu $\boldsymbol{K}C$, označujícího ty objekty, o nichž znalostní báze ví, že tvoří instance C. Potom je možno spouštěcí pravidlo $C \Rightarrow D$ převést na pravidlo inkluze konceptů $\boldsymbol{K}C \sqsubseteq D$.

Pravidlové znalostní báze sestávají z TBoxu, ABoxu a množiny pravidel typu $\boldsymbol{K}C \sqsubseteq D$.

9.4 ROZHODNUTELNOST, KOREKTNOST A ÚPLNOST

V odstavcích 9.3.1 a 9.3.3 byly uvedeny dvě z formálních metod, a to tablová metoda rozhodování splnitelnosti (množin) formulí a rozhodování subsumpce na základě strukturálního srovnávání. Nesplnitelnost množiny formulí je v základních jazycích DL, vybavených i dalšími potřebnými konstruktory, rozhodnutelná (tablový důkaz, který se na rozdíl od logiky prvního řádu vždy zastaví), sémanticky korektní a úplná (obr. 9.4).

Rozhodování subsumpce strukturálním srovnáváním je zřejmě konečným procesem (uvažují se konečné formule) rozhodnutelným s nejvýše polynomickou složitostí. Sémantická korektnost a úplnost formálního procesu vyplývá z jednoznačné korespondence subsumpce a odpovídající množinové inkluze.



obr. 9.4

9.5 VIZE BUDOVÁNÍ AGENTOVÝCH IS NA BÁZI DL

Předpokládejme, že se agent má pohybovat ve speciálním hostitelském prostředí za účelem získání informací pro určitého klienta. Agent i prostředí musí být vybudováni na bázi inferenčního servisu deskripční logiky. Lze předpokládat, že hledané informace budou v hostitelském prostředí součástí ABoxu nějaké znalostní báze. Znalosti formulované na konceptuální úrovni, potřebné pro získání hledaných informací, jsou formulovány v TBoxu odpovídající znalostní báze. Obsahy TBoxů a ABoxů příslušných znalostních bází lze přečíst z dokumentů RDF a DAML + OIL, uložených v agentově hostitelském prostředí. Dále stačí aplikovat některý z inferenčních systémů DL, jako je např. RACER.

Hostitelské prostředí může být vytvořeno pro speciální třídu agentů, tj. agent a prostředí sdílejí tutéž ontologii, nebo je ke komunikaci agenta s neznámým prostředím (založeným na odlišné ontologii) využit tzv. ontologický agent, který provádí překlady mezi ontologiemi.

Deskripční logika je axiómatickým systémem, proto rozhodování, resp. dokazování znalostí v DL systému je deduktivním procesem, vycházejícím ze znalostní báze Σ = (**T**, **A**), která představuje výchozí množinu speciálních axiómů. Pomocí formálních pravidel lze pak odvozovat ze znalostní báze další implicitně skryté znalosti.

Rozhodování *konceptové splnitelnosti*, tj. dokazování $\Sigma \not\models C \equiv \bot$ je problémem zjišťování existence modelu $\boldsymbol{I}$ znalostní báze Σ takového, v němž $C^I \neq \emptyset$. V jazycích s konstruktorem negace konceptu lze též problém *subsumpce* převést na konceptovou splnitelnost, neboť platí, že $\Sigma \models C \sqsubseteq D$, právě když $\Sigma \models C \sqcap \neg D \equiv \bot$.

Kontrolou instancí ($\Sigma \models C(\mathrm{a})$) se rozumí zjišťování, zdali tvrzení $C(\mathrm{a})$ je splněno v každém modelu Σ. V jazycích vybavených konstruktorem negace konceptu (komplementu) lze kontrolu instancí převést na problém konsistence znalostní báze, neboť platí $\Sigma \models C(\mathrm{a})$, kde a je individuum, právě když $\Sigma \cup \{\neg C(\mathrm{a})\} \models \bot$.

Základem odvozování ze znalostní báze Σ je zjišťování, zdali vztah (např. subsumpce) mezi dvěma konceptovými výrazy je logickým důsledkem jejich deklarací v Σ.

Formule φ je logickým důsledkem znalostní báze Σ, tj. $\Sigma \models \varphi$, je-li splněna ve všech modelech Σ. V jazycích vybavených konstruktorem negace konceptu (komplementu) důkaz logického důsledku převést na problém konsistence znalostní báze rozšířené o negaci dokazované formule, neboť platí $\Sigma \models \varphi$, právě když $\Sigma \cup \{\neg\varphi\} \models \perp$. Kontrola instancí je speciálním případem důkazu logického důsledku.

Tablový důkaz splnitelnosti konceptové formule nebo množiny konceptových formulí je formálním důkazem, který spočívá v aplikaci tablových pravidel, generujících jejich modely. Nemožnost nalezení modelu svědčí o nesplnitelnosti této množiny formulí. Nesplnitelnost množiny formulí je v základních jazycích DL, vybavených i dalšími potřebnými konstruktory, rozhodnutelná (tablový důkaz, který se na rozdíl od logiky prvního řádu vždy zastaví), sémanticky korektní a úplná.

Kontrolní otázky

1. Jak lze rozhodnout (sémanticky/formálně), zdali je koncept splnitelný ?
2. Kdy je koncept logickým důsledkem znalostní báze ?

Literatura

1. Baader,F. et coll. : The Description Logic Handbook. Cambridge University Press, 2004., ISBN 0521781760.
2. Baader, F., Kűsters, R., Borgida, A., McGuinness, D. L. : Matching in Description Logics.J. Logic. Computat., Vol. 9, No. 3, pp. 411-447, 1999.
3. Baader, F., Nutt, W. : Basic Description Logics. In Description Logic Handbook, Cambridge University Press, 2002, pp. 47 – 100.
4. Borgida, A. : On the relative expressiveness of description logics and predicate logics. Artificial Intelligence 82, 1996, pp 353-367.¨
5. Brachman, R.J, Levesque, H.J. : Readings in Knowledge Representation. Morgan Kaufmann, Los Altos, 1985.
6. Donini, F.M., Lenzerini, M., Nardi, D., Schaerf, A. Reasoning in Description Logic A Great Collection U. Gnowho and U. Gnowho-else, eds, CSLI Publications, 1997
7. Calvanese, D., Lenzerini, M.G., Nardi, D., : Description Logics for Conceptual Data Modeling. Logics for Databases and Information systems, J. Chomicki and G. Saake eds., Kluwer, 1998.
8. Calvanese, D., De Giacomo, G., Nardi, D., Lenzerini, M. : Reasoning in Expresive Description Logics. In : Handbook of automated reasoning, Ed. Alan Robinson and Andrei Voronikov, Elsevier Sci. Publishers B.V., 2000.
9. Franconi, E. : http://www.cs.man.ac.uk/~franconi
10. Ohlbach, H.J., Koehler, J. : Modal logics, description logics and arithmetic reasoning. Artificial Intelligence 109, 1999, pp 1-31.
11. Schaerf, A. : Reasoning with Individuals in Concept Languages. Data and Knowledge Engeneering, Vol. 13(2), pp 141-176.
12. Schmidt-Schau3, M., Smolka, G. : Attributive Concept Descriptions with Complements. Artificial Intelligence Journal 48(1), 1991, pp 1-26.

10 REPREZENTACE ZNALOSTÍ KONCEPTUÁLNÍMI GRAFY

V této lekci, jejíž prostudování by vám mělo trvat zhruba 1,5 h, se seznámíte s dalším jazykem reprezentace znalostí na konceptovém principu. Tímto jazykem je jazyk konceptuálních grafů J. W. Sowy, který je na rozdíl od jazyka deskripční logiky podobně jako jazyk asociativních sítí jazykem srozumitelným i laikům. Uvidíte, že znalostní bází může být i soubor grafů a jejich podgrafů, umožňujících zachytit ve znalostní bázi kontext znalostí.

10.1 KONCEPTUÁLNÍ GRAF

Konceptuální grafy definované zde podle J.W. Sowy [3] jsou schopny reprezentovat širokou oblast znalostí od predikátové logiky k přirozenému jazyku.

Definice 10.1 (konceptuálního grafu)

Konceptuální graf je síť uzlů dvou druhů pro reprezentaci

1. konceptů
2. jejich vzájemných vztahů

Hrany mezi těmito uzly jsou orientované. Konceptuální graf je bipartitní graf. Hrana nikdy nespojuje dva koncepty ani dva vztahy, vždy spojuje koncept se vztahem nebo naopak.

Konceptuální grafy využívají dva typy notace – grafickou a lineární.

Definice 10.2 (jazykových symbolů)

Jazykovými symboly grafické notace konceptuálních grafů jsou :

- obdélníky reprezentující v grafu koncepty a jejich individuové instance,
- elipsy reprezentující vztahy mezi nimi (role).

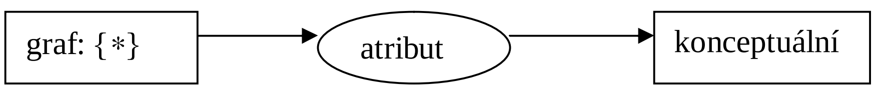
Lineárními jazykovými prostředky pro reprezentaci konceptuálních grafů jsou :

- koncepty uváděné v hranatých závorkách,
- vztahy uváděné v okrouhlých závorkách.

Koncepty, které reprezentují podgrafy jsou nástrojem vyjádření kontextu.

Příklad 10.1

Graf reprezentující pojem konceptuálního grafu :



Lineární notace :
[Graf: {*}] ⟶ (Attr) ⟶ [konceptuální]

Speciální jména jsou přiřazena těmto druhům konceptu :

1. blank je prázdný konceptuální graf
2. singleton je samostatně stojící koncept
3. hvězdice je konceptuální graf, který sestává z jednoho vztahu a připojených konceptů.

Příklad 10.2

Reprezentace následujících tvrzení lineární formou:

„Pták zpívá."
[Zpívat] ⟶ (Agnt) ⟶ [Pták]
„Jan jede do Aalborgu."
[osoba: Jan] ⟵ (Agnt) ⟵ [Jet] ⟶ (Dest) ⟶ [Město : Aalborg]
Jet má agenta, kterým je osoba Jan, místo určení, kterým je město Aalborg.
„Kniha je na stole."
[kniha] ⟶ (on) ⟶ [stůl]

Definice 10.3 (konceptu v systému konceptuálních grafů)

Koncept sestává ze dvou složek :

1. typ konceptu
2. referent

označení v lineární notaci : [typ : referent]

Např. výraz [osoba: Jan] reprezentuje tvrzení „(existuje) osoba jménem Jan"
Referent může být i prázdný – např. výraz [kniha] reprezentuje tvrzení „existuje kniha"

Definice 10.4 (vztahu v systému konceptuálních grafů)

Se vztahy souvisejí tři pojmy : *typ vztahu*, *valence* a *signatura*

Typem vztahu je jméno vztahu. Každý vztah má typ vztahu (nemusí mít referenta). Typ vztahu určuje valenci a signaturu, tj. počet hran k němu příslušných a typy konceptů, které se k němu vztahují.

Valence vztahu je konstanta označující počet hran patřících k danému vztahu. Vztah valence n se nazývá n-adický vztah, speciálně monadický (n = 1), dyadický (n = 2), triadický (n = 3).

Signatura vztahu je seznam typů konceptů patřících ke vztahu. Signatura vztahu se zapisuje jako seznam $<t_1,\ldots t_n>$ typů konceptů připojených ke vztahu.

Příklad 10.3

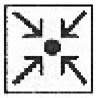
Některá standardní jména vztahů podle J.W. Sowy :

On

In
Dest (destination)
Agnt (agent)
Thme (thema)
Ptnt (patient)
Rcpt (recipient)

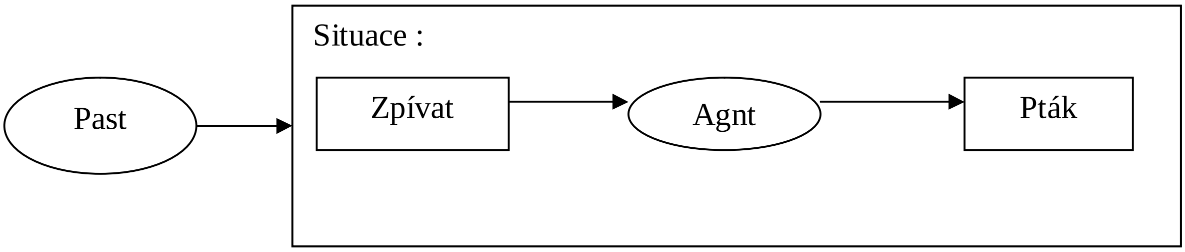
Příklad využití : „Pták zpívá na jabloni.“
[Zpívat] ⟶ (Agnt) ⟶ [Pták] ⟶ (In) ⟶ [Jabloň]
Typ vztahu Agnt je Agnt, valence je 2 a znamená vztah mezi činností a živou bytostí.

Příklad 10.4

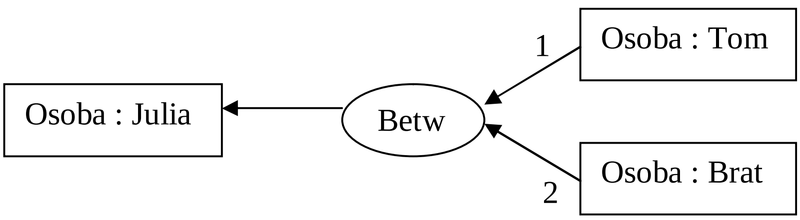


Speciální případy vztahů :

Monadický vztah je např. (Past) – pro minulý čas, podobně (Psbl) – pro možná.
(Past) ⟶[Situace : [Zpívat] ⟶ (Agnt) ⟶ [Pták]]



Příklad *triadického vztahu* :
[Osoba : Julia] ⟵ (Betw) ←1— [Osoba : Tom]
←2— [Osoba : Brad]
„Julie je mezi Tomem a Janem.“



Definice 10.5 (konvence směru hran)

Konvence směru hran: Jestliže vztahu náleží n hran, směřuje prvních n-1 hran ke vztahu, zbývající n-tá hrana od vztahu. Toto pořadí je dáno signaturou.

10.2 ZNALOSTNÍ BÁZE

Definice 10.6

Znalostní báze v systému konceptuálních garů sestává z těchto složek :

- Soubor konceptuálních grafů vyjadřujících znalosti o modelovaném světě - referenčním systému.
- Hierarchie typů vyjadřující typy konceptů, potřebných k reprezentaci daného referenčního systému.
- Hierarchie typů vztahů, vyjadřující typy vztahů, potřebných k reprezentaci daného referenčního systému.
- Katalog objektů, který obsahuje identity všech objektů, které se objeví ve znalostní bázi.

10.2.1 Ontologie jako základní slovník konceptuálních grafů

J.W. Sowa definuje ontologii (viz též lekci 7) takto : „ Předmětem ontologie je studium kategorií věcí, které existují nebo mohou existovat v nějaké oblasti. Produkt takového studia, zvaný ontologie, je katalog typů věcí považované za existující v oblasti zájmu D z pohledu osoby, která používá jazyk L pro účely popisu oblasti D.“

Základním pojmem ontologie jako principu konceptuálních grafů je pojem typu. V konceptuálních grafech vystupují *typy konceptů* a *typy vztahů*.

Definice 10.7

Typ je návěští (jméno), které je přiřazeno množině entit podobného pojetí.

Těmito entitami mohou být např.
Individuály (jako osoba, kočka, autobus,…)
Vlastnosti (jako krásný, inteligentní, smrtelný, modrý,…)
Akce (jako políbit, bliknout, jít,..)
Abstraktní entity (jako PřirozenéČíslo, Množina,…)

Přestože typ je jméno skupiny entit, vždy existuje pouze jako abstraktní. Obráceně neplatí, že cokoliv abstraktního je typem. Např. 2 je abstrakní, ale není typem, nýbrž instancí typu PřirozenéČíslo.

Definice 10.8 (způsobů definování typů)

Typy mohou být definovány následujícími způsoby :

- Extenzí – např. typ RodinaKovářů je dána výčtem <Daniela, Jiří, Jana, Jan>
- Intenzí – seznamem vlastností individuí náležících typu, resp. obecným pravidlem řazení individuí k tomuto typu (typ LichéČíslo je číslo nedělitelné dvěma)
- Axiómem – typy, které jsou matematickými entitami. Např. Peanovy axiómy definující typ PřirozenéČíslo :
  1. 0 je přirozené číslo
  2. Je-li a přirozené číslo, je a+1 rovněž přirozené číslo.
  3. Jestliže z dokazatelnosti nějaké vlastnosti pro a vyplývá její dokazatelnost pro a+1 a

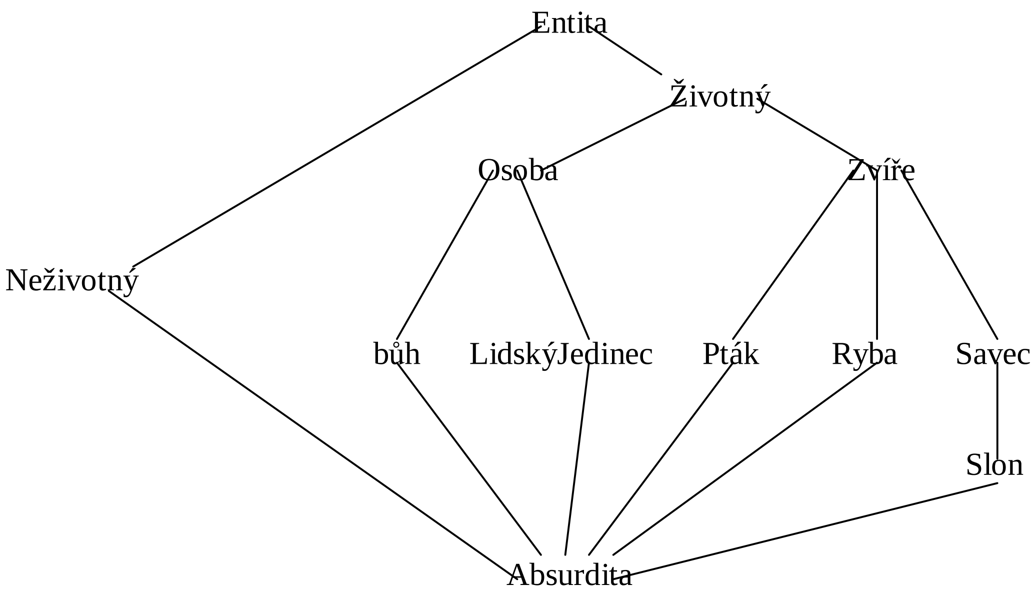
jestliže je tato vlastnost dokazatelná pro 0, pak tato vlastnost platí pro všechna přirozená čísla.
4. Jestliže a+1 = b+1, potom a = b.
5. Přičtením 1 k přirozenému číslu nelze nikdy získat číslo 0.
Čísla, která splňují axiómy 1. – 5. jsou právě přirozená čísla.

- Odkazem (referencí) na jiný typ s diferenciací – StálýZaměstnanec je Zaměstnanec se stálou smlouvou. – Poznámka : i to jsou intenze.

Typy lze organizovat do *hierarchií*. Hierarchie typů tvoří svaz (viz následující lekci), v němž je částečné uspořádání dáno relací subtypu (označuje se A $\subseteq$ B, je-li A subtypem B – např. pták $\subseteq$ živočich).

Příklad 10.5

Příklad ontologie :



obr. 10.1

….

10.2.2 Relace subtypu

Definice 10.9

A je subtypem B, tj. A $\subseteq$ B, jestliže

- buď A je B
- nebo A je specializací B

Např. typ A (Pták) je specializací typu B (Zvíře).
Specializace A jiného typu B zahrnuje všechny vlastnosti B s přidáním některých dalších omezení – např. „má křídla“, „klade vejce“,…
Pro relaci subtypu platí, že je tranzitivní.

10.2.3 Entita a absurdita

Definice 10.10

Entita (top) je univerzální typ – typ, který je supertypem všech ostatních typů v hierarchii. Typ Entita reprezentuje libovolnou entitu.

Absurdita (bottom) je absurdní typ – typ, který je subtypem všech ostatních typů v hierarchii.

10.2.4 Dědičnost

Je-li A $\subseteq$ B, A *dědí* všechny vlastnosti B, tedy cokoliv platí o B, platí též o A. Savci dědí všechny vlastnosti zvířete.

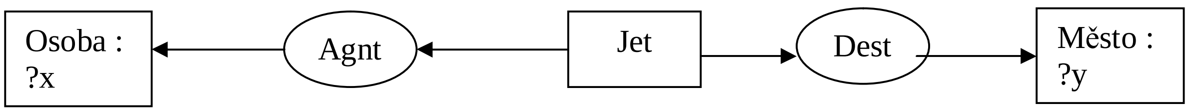
Částečné uspořádání subtypu je reflexivní, antisymetrické a tranzitivní (viz následující lekce).

10.2.5 Lambda výrazy pro definování konceptů a vztahů

Jde o výrazy, které umožňují formulovat obecná tvrzení.

Příklad 10.6

„Někdo jede do města."
[Osoba : ?x] ⟵ (Agnt) ⟵ [Jet] ⟶ (Dest) ⟶ [Město : ?y]



Koncept [Profesor : Alfred] je aktuálním parametrem uvedeného lambda výrazu
[Profesor : Alfred] ⟵ ([Osoba : ?x] ⟵ (Agnt) ⟵ [Jet] ⟶ (Dest) ⟶ [Město : Aalborg])
Profesor je subtypem Osoba
[Profesor : Alfred] ⟵ (Agnt) ⟵ [Jet] ⟶ (Dest) ⟶ [Město : Aalborg]

10.3 ONTOLOGIE V KONCEPTUÁLNÍCH GRAFECH

10.3.1 Typy konceptů a typy vztahů

Každá znalostní báze musí mít hierarchii konceptů T a každý její konceptuální graf se musí vztahovat k T.

Definice 10.11 (hierarchie typů konceptů)

Hierarchie typů konceptů T je síť (svaz) návěští typů, která mohou být dvojího typu :

- Primitivní, která jsou pouhými jmény zařaditelnými do hierarchie T
- Definovaná pomocí monadických lambda výrazů

Každá hierarchie typů obsahuje dvě primitivní typová návěští – entitu a absurditu.

Všechny koncepty mají typ konceptu. Pro každý koncept je typem konceptu buď návěští z T nebo monadický lambda výraz. To znamená, že v hierarchii typů není třeba všech možných typů. Ty se dodefinují monadickým lambda výrazem.
Typ konceptu nemůže zůstat na rozdíl od referenta prázdný, např. [: Jan].
Typem konceptu může být

- návěští typu z T
- monadický lambda výraz.

Definice 10.12 (hierarchie typů vztahů)

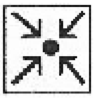
Hierarchie typů vztahů R je síť (svaz) návěští typů, která mohou být dvojího typu :

- Primitivní, která jsou pouhými jmény zařaditelnými do hierarchie R
- Definovaná pomocí n-adických lambda výrazů

Primitivní návěští typu vztahu jsou zařaditelná do R.
Definovaná návěští typů vztahů sestávají z

- jména
- n-adického lambda výrazu.

Příklad 10.7



Vztah JeModrý :
[Hračka : Míč]) ⟵ ([Předmět : ?x] ⟶ (Attr) ⟶ [Modrý])

n-adické návěští vztahu může být subtypem pouze opět n-adického návěští vztahu (stejné valence).

10.3.2 Referenty

Nejdůležitější druhy referentů :

- **Prázdný referent** : např. [Auto] - „existuje auto“, [Auto : Fabia] „existuje auto jménem Fabia“.
- **Lokátor** : individuální označení - pojmenovaný referent
  Lokátor je referent, který říká, kde lze nalézt individuum, k němuž směřuje reference – fyzický svět, katalog individuí.
  Existují dva druhy lokátorů
  1. individuální označovače
  Individuálními označovači jsou např. jména osob, míst, organizací. Např.
  [Osoba : Jan] ⟵ (Expr) ⟵ [Žít] ⟶ (Loc) ⟶ [Město : Aalborg]
  “Jan žije v Aalborgu“. Zde jsou lokátory Jan a Aalborg, které jsou individuálními označeními.

2. indexace
Indexační referenty jsou používány, je-li třeba referovat na individua, jejichž identita závisí na kontextu. Zde # označuje „určitý“. Např. [Mléko : #] ⟵ (Thme) ⟵ [Pít] ⟶ (Agnt) ⟶ [Kočka : #]. „Určitá kočka pije určité mléko“.

- **Množina věcí** :
  [Host : {Tom, Julia, Brad}] ⟵ (Agnt) ⟵ [Zpívat] ⟶ (Thme) ⟶ [Píseň : HappyBerthdayToYou] ⟶ (To) ⟶ [Osoba : Alfred]
  Hosté Tom, Julia a Brad zpívají Alfrédovi píseň Happy Birthday To You.
  Speciální množinou je {*} - např. [Pták : *]
  [Zpívat] ⟶ (Agnt) ⟶ [Pták : *] ⟶ (In) ⟶ [Třešeň] – „Ptáci zpívající na třešni“.
- **Deskriptory** : Celé grafy mohou být referenty.
  (Past) ⟶ [Situace : [Osoba : Jan] ⟵ (Agnt) ⟵ [Políbit] ⟶ (Benf) ⟶ [Osoba : Mary]].
  Zde konceptuální graf [Osoba : Jan] ⟵ (Agnt) ⟵ [Políbit] ⟶ (Benf) ⟶ [Osoba : Mary] je referentem konceptu typu Situace. Zde graf vyplňuje slot referenta.
- **Počty** : Pomocí symbolu @ lze stanovit, kolika individuí se referent týká.
  [Osoba] ⟶ (Has) ⟶ [Noha : @2]
- **Univerzální kvantifikace** – Př. [ŽijícíRyba : ∀] ⟶ (Attr) ⟶ [Mokrý]
  “Všechny žijící ryby jsou mokré“.

Definice 10.13 (referenta)

Koncept má typ a referenta : [typ : referent]
Referent sestává z

- jeho kvantifikátoru
- jeho designátoru

Kvantifikátory mohou být

- Existenční kvantifikátor : je reprezentován buď prázdným referentem nebo symbolem ∃ . Např. [Kočka] ⟶ (On) ⟶ [Matrace]. Nějaká kočka je na nějaké matraci.
- Definovaný kvantifikátor : viz např. univerzální kvantifikaci nebo množinu výčtem prvků nebo množinu s * (viz výše).

Designátory mohou být

- **Literál** je syntaktický reprezentant formy referentu. Jsou tři typy literálů :
  1. číslo - [Číslo : 18] (rozlišit od kvantifikátoru @18, který udává počet)
  2. string - [String : abcd],
  3. kódovaný literál - [Míra : <18, cm>].
- **Lokátor** indikuje způsob vyhledání referenta - [Kočka : Micka], [Student : #ty] - (viz výše)
- **Deskriptor** – konceptuální graf je deskriptorem referenta (viz výše)

10.3.3 Zahnízděné grafy

Definice 10.14 (zahnízděného grafu)

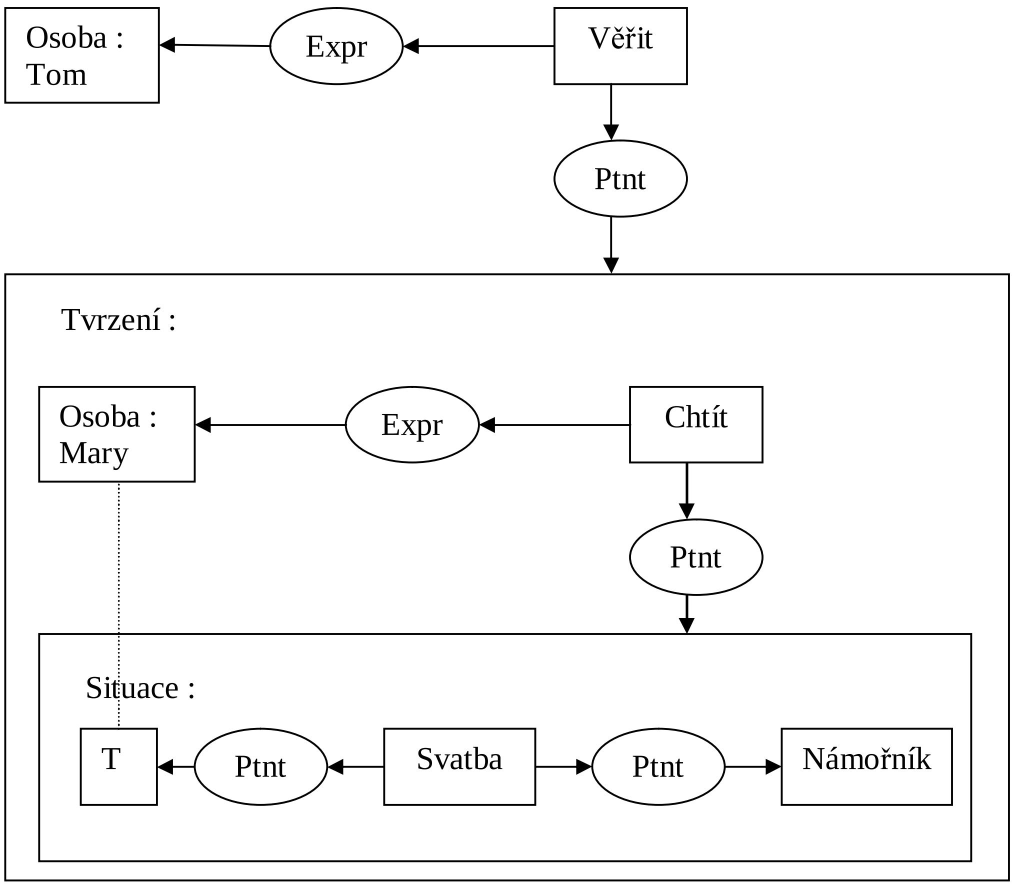
Graf G1 je zahnízděn v konceptu C, jestliže
buď G1 je přímou součástí referenta C
nebo G1 je přímou částí referenta konceptu C2, který je zahnízděn v C.

Příklad 10.8 (Sowa, 1984) :

Na obr. 10.1 je reprezentováno toto tvrzení :
„Tom věří, že si Mary chce vzít námořníka.“

Lineární forma reprezentace uvedeného konceptuálního grafu na obr. 10.1 :

```
[věřit] –
  (Expr) ⟶ [Osoba : Tom]
  (Ptnt) ⟶ [Tvrzení :
              [Chtít] –
                (Expr) ⟶ [Osoba : Mary]
                (Ptnt) ⟶ [Situace :
                   [Mary] –
                     (Agnt) ⟶ [T : Mary]
                     (Ptnt) ⟶ [Námořník]
              ]
           ]
```



obr. 10.2

10.3.4 Kontext

Definice 10.15 (kontextu)

Kontext C je koncept, jehož designátorem je neprázdný konceptuální graf G.

V předcházejícím příkladě tvoří tvrzení kontext, v němž se vyskytuje subgraf.

10.4 KONCEPTUÁLNÍ GRAF A LOGIKA

Každý konceptuální graf lze převést do formule predikátové logiky.

10.4.1 Negace v konceptuálním grafu

V konceptuálním grafu se symbol negace umisťuje výhradně před kontext (Situace, Předpoklad..).

Příklad 10.9

Negace v grafu :

¬[Situace : [Slunce : #] ⟵ (Agnt) ⟵ [Svítit]]
„Slunce nesvítí".

¬[Předpoklad : [Muž] ⟵ (Expr) ⟵ [Rozumět] ⟶ (Thme) ⟶ [Žena : {*}]]
„Není pravda, že existuje muž, který rozumí ženám".

10.4.2 Konjunkce v konceptuálním grafu

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

Příklad 10.10

Konjunkce v grafu :

[Předpoklad : [Žena : *x] ⟶ (Attr) ⟶ [Krásná]
[Žena : *x] ⟶ (Attr) ⟶ [Nebezpečná]]
„Existuje žena, která je krásná a nebezpečná".

10.4.3 Disjunkce v konceptuálním grafu

V konceptuálním grafu se disjunkce G1 ∨ G2 řeší pomocí konjunkce a negace – DeMorganovým pravidlem :

1. Negace kontextu přes G1
2. Negace kontextu přes G2
3. Zařazení 1. a 2. do společného kontextu
4. Negace 3.

Příklad 10.11

Řešení disjunkce DeMorganovým pravidlem :

¬[Situace :
    ¬[Situace :
        [Osoba : Jan]- - -[Blázen]
    ]
¬[Situace :
        [Osoba : Jan] ⟶ (Attr) ⟶ [Chytrý] ⟶ (Meas) ⟶ [Stupeň : #velmi]
  ]
 ]

10.5 VÝBĚR KONCEPTUÁLNÍCH VZTAHŮ (ROLÍ) PODLE SOWY

Příklad 10.12

Nejužívanější typy vztahů na příkladech :

Agnt (Act, Animate) – Agent : Aktivní živočich, který vyvolá akci.
[ŘíditAuto] ⟶ (Agnt) ⟶ [Osoba : Alfred]

Attr (Object, Entity) – Attribute : Entita, která je atributem objektu.
[Papoušek : Figo] ⟶ (Attr) ⟶ [Andulka]

Benf (Act, Animate) – Beneficiary : Příjemce akce, který též řídí a má užitek z úspěšného provedení akce.
[Dar : {*}] ⟵ (Thme) ⟵ [Dát] ⟶ (Benf) ⟶ [Osoba : Alfréd]
„Dar byl dán Alfrédovi“.

Chrc (Entity, Entity) - Characteristic : Typ, jehož instancemi jsou vlastnosti entit
[Papoušek : Figo] ⟶ (Chrc) ⟶ [Modrý]

Cmpl (TemporalProcess, Physical) – Completion : Cíl dokončeného procesu, jehož hlavním rysem je, že probíhá v čase. Tím může být např. stav anebo situace.
[Šaty] ⟵ (Ptnt) ⟵ [Proces : Sušit] ⟶ (Cmpl) ⟶ [Stav : suchý]
„Šaty se suší, až jsou suché".

Dest (SpatialProcess, Physical) – Destination : Cíl procesu, jehož hlavním rysem je, že je v prostoru.
[Osoba : Romeo] ⟵ (Agnt) ⟵ [Jet] ⟶ (Dest) ⟶ [Město : Mantova]

Dur (State, Interval) – Duration : Interval, v němž stav trvá.
[Film] ⟵ (Thme) ⟵ [Předvádění] ⟶ (Dur) ⟶ [Interval : @120 min]

Efct (Entity, Entity) – Effector : Aktivní entita (živá nebo neživá), která vyvolá akci bez vlastního záměru.
[Dobrý] ⟵ (Attr) ⟵ [Strom] ⟵ (Efct) ⟵ [Produkovat] ⟶ (RSLT) ⟶ [Ovoce] ⟶ (Attr) ⟶ [Hodně]
„Dobrý strom produkuje hodně ovoce".
Expr (State, Animate) – Experiencer : Živá entita, která zakouší nějaký stav.
[Osoba : Romeo] ⟵ (Expr) ⟵ [Slyšet] ⟶ (Thme) ⟶ [Věta : ‚Ach!']

Role; Has (Entity, Entity) – Role : Jde o primitivní vztah – nedef.
[Osoba : ∀] ⟵ (Has) ⟶ [Část : @2 nohy]

Ins (Act, Entity) – Instrument . Jde o nástroj použitý v akci.
[Nůž] ⟵ (Inst) ⟵ [ZabítSe] ⟶ (Agnt) ⟶ [Osoba : Julie]

Loc (Physical. Physical) – Location : Místo, kde něco je, koná se.
[Osoba : Princ] ⟵ (Agnt) ⟵ [Přijít] ⟶ (Loc) ⟶ [Místo : Kobka]

Manr (Process, Entity) - Manner : Entita, která je vlastností nějakého procesu.
[Vědec : Albert] ⟵ (Agnt) ⟵ [Navrhnout] ⟶ (Thme) ⟶ [Definice : @ 2.3] ⟶ (Manr) ⟶ [Zkusmo]

Meas (Attribute, Quantity) – Measure : Kvantita popisující atribut.
[Osoba : Jan] ⟶ (Attr) ⟶ [Chytrý] ⟶ (Meas) ⟶ [Stupeň : #velmi]

Part (Object, Object) – Part : Objekt, který je částí druhého objektu.
[Tělo] ⟶ (Part) ⟶ [Oko]

Path (Process, Place) – Path : Cesta popisující proces.
[Obchodník : #] ⟵ (Agnt) ⟵ [Cestovat] ⟶ (Path) ⟶ [Město : {Praha, Olomouc, Ostrava}]

Poss (Animate, Entity) – Possesion : Entita vlastněná nějakou živou bytostí.
[Auto : *x] ⟵ (Thme) ⟵ [Řídit] ⟶ (Agnt) ⟶ [Osoba : Alfred] ⟶ (Poss) ⟶ [Auto : ?x]
„Alfréd řídí své auto".

Ptim (Physical, Time) – PointIn Time : Hlavní participant časového společenství (nexus).

[Čas : 5.13 am] ⟵ (PTim) ⟵ [Situace : [Osoba : Romeo] ⟵ (Agnt) ⟵ [Odejít]]

Ptnt (Process, Physical) – Patient : Participant procesu, který podstoupí změnu během procesu.
[Nůž] ⟵ (Inst) ⟵ [Zabít] ⟶ (Agnt) ⟶ [Osoba : Julie *x] ⟶ (Ptnt) ⟶ [Osoba : ?x]
„Julie se zabila nožem".

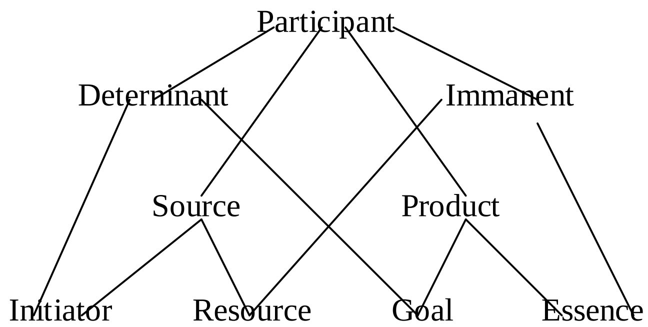
Rcpt (Act, Animate) – Recipient : Životný cíl akce.
[Osoba : Julie] ⟵ (Rcpt) ⟵ [Dát] ⟶ (Agnt) ⟶ [Osoba : Romeo] ⟶ (Thme) ⟶ [Polibek]

Rslt (Process, Entity) – Result : Neživotný cíl akce.
[Osoba : Jan] ⟵ (Agnt) ⟵ [Uvařit] ⟶ (Rslt) ⟶ [Čaj : #42]
„Jan uvařil čaj č.42".

Thme (Situation, Entity) – Theme : Participant řekl nebo zažil, ale nezměnil.
[Osoba : Julie] ⟵ (Agnt) ⟵ [Říci] ⟶ (Thme) ⟶ [Věta : „Je konec"]

10.6 TÉMATICKÉ ROLE

Konceptuální vztah (role) spojuje koncept slovesa (procesu) s konceptem participanta. V rámcové reprezentaci se jedná o sloty. V ontologii podle Sowy jsou tématické role klasifikovány do subtypů typu Participant.



obr. 10.3

Determinant determinuje směr procesu, buď od počátku (iniciátora) nebo vzhledem k cíli.

Immanent je reprezentován procesem, ale není zde aktivní kontrola, jak dopadne.

Source musí být přítomen na začátku procesu, ale nemusí nutně participovat během procesu.

Product musí být přítomen na počátku procesu, ale nemusí nutně participovat během procesu.

Např. "Zuzana posílá Bobovi dárek Expresem". Dárek (Essence) a Expres (Resource) jsou immanentní participanti, protože jsou přítomny od začátku do konce procesu od iniciátora (Zuzana) po cíl (Bob).

V souladu s Aristotelovým pojetím rozlišuje Sowa příčiny dění :
Initiator – když je zahájena změna nebo stav
Resource – odpovídá materiální příčině
Goal – jde o účel nebo prospěch
Essence – odpovídá formální příčině, která je esencí dění.
Na základě nich charakterizuje Sowa participanty akcí (odlišně od Aristotela) :

	Initiator	Resource	Goal	Essence
Action	Agent,Effector	Instrument	Result, Recipient	Patient, Theme
Process	Agent, Origin	Matter	Result, Recipient	Patient, Theme
Transfer	Agent, Origin	Instrument,Medium	Experiencer,Recipient	Theme
Spatial	Origin	Path	Destination	Location
Temporal	Start	Duration	Completion	PointInTime
Ambient	Origin	Instrument,Matter	Result	Theme

tab. 10.1

Tab. 10.1 je důležitým východiskem, ale vyžaduje další rozlišování. Např. „Tom upekl koláč.“ – zde koláč může být result (determinant product) nebo patient (Immanent product), který byl zahříván. Tomu odpovídají dva grafy :
[Osoba: Tom] ← (Agnt) ← [Péci] → (Rslt) → [koláč: #]
[Osoba: Tom] ← (Agnt) ← [Péci] → (Ptnt) → [koláč: #]
Podle hierarchie participantů ale je Result < Goal < Product a Patient < Essence < Product. To vede ke zjednodušení grafu
[Osoba: Tom] ← (Agnt) ← [Péci] → (Prod) → [koláč: #]

Podobně „Pes rozbil okno.“ může znamenat určené nebo náhodné rozbití, resp. jako prostředek, aby se dostal na druhou stranu. Tomu odpovídají grafy :
[Pes: #] ← (Agnt) ← [Rozbít] → (Ptnt) → [Okno: #]
[Pes: #] ← (Efct) ← [Rozbít] → (Ptnt) → [Okno: #]
[Pes: #] ← (Inst) ← [Rozbít] → (Ptnt) → [Okno: #]
Protože Agent < Initiator < Source a Effector < Initiator < Source a Instrument ⊆ Resource ⊆ Source, lze zjednodušit
[Pes: #] ← (Srce) ← [Rozbít] → (Ptnt) → [Okno: #]

Popis tématických rolí uvedených v tabulce 10.1:

Agnt ⊆ Initiator : Agnt(Act, Animate)
Aktivní životná entita, která z vlastní vůle iniciuje akci.
„Eva utrhla jablko.“
[Osoba: Eva] ← (Agnt) ← [Utrhnout] → (Ptnt) → [Jablko: #]

Beneficiary ⊆ Recipient : Benf(Act, Animate)
Příjemce, který odvodí zisk z úspěšného dokončení akce.
„Diamant byl dán Ruby.“
[Diamant : {*}] ← (Thme) ← [Dát] → (Bnft) → [Osoba : Ruby]

Completion ⊆ Goal : Cmpl(TemporalProcess, Physical)
Cíl procesu v čase.

„Mary čekala do noci.“
[Osoba: Mary] ← (Thme) ← [Čekat] → (Cmpl) → [Noc]

Destination ⊆ Goal : Dest(SpatialProcess, Physical)
Cíl procesu v prostoru.
„Bob jel do Prahy.“
[Osoba: Bob] ← (Agnt) ← [Jet] → (Dest) → [Město : Praha]

Duration ⊆ Resource : Dur(State, Interval)
Resource procesu v čase.
„Auto bylo opravováno 5 hodin.“
[Auto : #] ← (Thme) ← [Opravovat] → (Dur) → [Interval : @5hod]

Effector ⊆ Initiator : Efct(Entity, Entity)
Aktivní determinantní zdroj, životný nebo neživotný, který iniciuje akci, ale bez vlastního záměru.
„Strom vytvořil nové listy.“
[Strom : #] ← (Efct) ← [Vytvořit] → (Rslt) → [List : {∗}] → (Attr) → [Nový]

Experiencer ⊆ Goal : Expr(State, Animate)
Aktivní životný cíl zkušenosti.
„Yojo vidí rybu.“
[Kočka : Yojo] ← (Expr) ← [Vidět] → (Thme) → [Ryba : #]

Instrument ⊆ Resource : Inst(Act, Entity)
Zdroj, který událost nezmění.
„Klíč otevřel dveře.“
[Klíč : #] ← (Thme) ← [Otevřít] → (Thme) → [Dveře : #]

Location ⊆ Essence : Loc(Physical, Physical)
Základní participant prostorového nexu.
„Vozy přijíždějí na stadion.“
[Vůz : {∗}] ← (Thme) ← [Přijíždět] → (Loc) → [Stadion]

Matter ⊆ Resource : Matr(Act, Substance)
Zdroj, který je událostí změněn.

Medium ⊆ Resource : Med(Transfer, Physical)
Fyzikální zdroj přenosu informace.
„Bill volá Borise telefonem.“
[Osoba: Bill] ← (Agnt) ← [Volat]
(Expr) → [Osoba : Boris]
(Med) → [Telefon]

Origin ⊆ Initiator : Orgn(Process, Physical)
Pasivní determinantní zdroj prostorového nebo ambient nexu.
„Kapitola začíná na stránce 20.“
[Kapitola : #] ← (Thme) ← [Začínat] → (Orgn) → [Stránka : 20]

Path ⊆ Resource : Path(Process, Place)
Zdroj prostorového nexu.
„Pizza byla převezena přes Albany a Bufallo.“
[Pizza : #] ← (Thme) ← [Převézt] → (Path) → [Město : {Albany, Bufallo}]

Patient ⊆ Essence : Ptnt(Process, Physical)
Hlavní participant, který prodělá nějaké strukturální změny jako výsledek události.
„Kočka sežrala kanára.“
[Kočka : #] ← (Agnt) ← [Sežrat] → (Ptnt) → [Kanár : #]

PointTime ⊆ Essence : Ptim(Physical, Time)
Esenciální participant temporálního nexu.
„V 5.30 Erin odjela.“
[Time : 5.30] ← (PTim) ← [Situation : [Osoba : Erin] ← (Agnt) ← [Odjet]]

Recipient ⊆ Goal : Rcpt(Act, Animate)
Životný cíl akce.
„Zuzana poslala dárek Bobovi.“
[Osoba : Zuzana] ← (Agnt) ← [Poslat] -
(Thme) → [Dárek : #]
(Rcpt) → [Osoba : Bob]

Result ⊆ Goal : Rslt(Process, Entity)
Neživotný cíl akce.
„Erik postavil dům.“
[Osoba : Erik] ← (Agnt) ← [Postavit] → (Rslt) → [Dům : #]

Start ⊆ Initiator : Strt(Entity, Time)
Determinovaný zdroj temporálního nexu.
„Bill čekal od poledne do tří.“
[Osoba : Bill] ← (Thme) ← [Čekat] -
(Strt) → [Poledne]
(Cmpl) → [Time : 3h]

Theme ⊆ Essence : Thme(Situation, Entity)
Esenciální participant, který má být pohybován, řečen, zkoušen, ale bez strukturální změny.
„Bill má rád pivo.“
[Osoba : Bill] ← (Expr) ← [MítRád] → (Thme) → [Pivo : #]

Konceptuální graf je síť uzlů dvou druhů pro reprezentaci konceptů a jejich vzájemných vztahů. Hrany mezi těmito uzly jsou orientované. Konceptuální graf je bipartitní graf. Hrana nikdy nespojuje dva koncepty ani dva vztahy, vždy spojuje koncept se vztahem nebo naopak. Konceptuální grafy využívají dva typy notace – grafickou a lineární.

Jazykovými symboly grafické notace konceptuálních grafů jsou :
- obdélníky reprezentující v grafu koncepty a jejich individuové instance,
- elipsy reprezentující vztahy mezi nimi (role).

Lineárními jazykovými prostředky pro reprezentaci konceptuálních grafů jsou :
- koncepty uváděné v hranatých závorkách,
- vztahy uváděné v okrouhlých závorkách.

Koncepty, které reprezentují podgrafy jsou nástrojem vyjádření kontextu.

Koncept sestává ze dvou složek, kterými jsou typ konceptu a referent (označení v lineární notaci : [typ : referent]).

Se vztahy souvisejí tři pojmy : *typ vztahu*, *valence* a *signatura*

Typem vztahu je jméno vztahu. Každý vztah má typ vztahu (nemusí mít referenta). Typ vztahu určuje valenci a signaturu, tj. počet hran k němu příslušných a typy konceptů, které se k němu vztahují.

Valence vztahu je konstanta označující počet hran patřících k danému vztahu. Vztah valence n se nazývá n-adický vztah, speciálně monadický (n = 1), dyadický (n = 2), triadický (n = 3).

Signatura vztahu je seznam typů konceptů patřících ke vztahu. Signatura vztahu se zapisuje jako seznam $<t_1,\ldots t_n>$ typů konceptů připojených ke vztahu.

Znalostní báze v systému konceptuálních grafů sestává z těchto složek :
- soubor konceptuálních grafů vyjadřujících znalosti o modelovaném světě -referenčním systému.
- hierarchie typů vyjadřující typy konceptů, potřebných k reprezentaci daného referenčního systému.
- hierarchie typů vztahů, vyjadřující typy vztahů, potřebných k reprezentaci daného referenčního systému.
- katalog objektů, který obsahuje identity všech objektů, které se objeví ve znalostní bázi.

Úkol 10.1

1. Vytvořte grafickou formu reprezentace tvrzení z příkladu 10.12.
2. Využijte konceptuálních vztahů z příkladu 10.13 k reprezentaci těchto tvrzení :

“Jan je lehce nachlazen.“
“Eva jede do Prahy.“
“Rek má dlouhou srst.“
“Honza prodává svůj dům.“

Kontrolní otázky

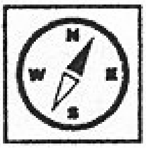
Jakými jazykovými symboly disponuje systém konceptuálních gtafů a) pro lineární vyjádření, b) pro grafické vyjádření ?
Co znamená, že konceptuální graf je bipartitní ?

Literatura

1. Kremer, R. : Visual Languages for Knowledge Representation.Proc. of KAW’98, Eleventh Workshop on Knowledge Aquisition, Voyager Inn, Banff, Alberta, Canada, 1998.
2. http://www.hum.auc.dk/cg/

3. Sowa, J.F. „ Knowledge Representation „ Logical, Philosophical, and Computational Foundations,
Brooks Cole Publishing Co., Pacific Growe, CA, 2000, 0-534-94965-7

11 FORMÁLNÍ KONCEPTOVÁ ANALÝZA



V předcházejících kapitolách jste se zabývali pojmem "koncept", přičemž intuitivní pochopení jeho významu vám zřejmě nečinilo potíže. Pokud jste si ale položili otázku, jak by bylo možno koncepty uchopit matematickými prostředky, jistě si se zájmem prostudujete tuto lekci, jejímž cílem je seznámit vás s formálními prostředky, na nichž je teorie konceptu vybudována, tj. matematickou oblastí zvanou formální konceptová analýza (FCA). Na praktických příkladech se zde seznámíte se základními pojmy formální koncepové analýzy, týkajícími se analýzované množiny objektů a atributů, kterými se tyto objekty vyznačují. Naučíte se na základě tabulky, představující formální kontext, v jehož rámci se analýza děje, vytvořit grafickou reprezentaci vztahů mezi objekty a jejich atributy, která tvoří konceptový svaz.

Prostudování této lekce by vám mělo trvat zhruba 2 h.

11.1 FORMÁLNÍ KONCEPTY, MATEMATIKA A LOGIKA

Stručně lze vyjádřit vztah mezi formální logikou a formální teorií konceptů takto :

Koncepty charakterizují entity modelovaného světa – referenčního systému. Koncept je, jak již bylo diskutováno v lekci 7, definován svou *intenzí* a *extenzí*. Intenzí se přitom rozumí formálně vyjádřený informační obsah potřebný k rozlišení, zdali objekt patří-do extenze (interpretující relace) konceptu. Z hlediska formální logiky intenzi konceptu lze charakterizovat jako logickou teorii. Extenze se stává modelem této teorie.

Formální konceptová analýza (FCA) je analýzou určitým způsobem (zpravidla tabulkou, tj. relací) explicitně daných informací (dat), která se provádí za účelem nalezení konceptů, jejich atributů a vzájemných vztahů v těchto datech. Data jsou zde formálními prostředky strukturována do jednotek, které by měly představovat formální abstrakce pojmů (konceptů) v lidské mysli, umožňující smysluplnou a úplnou interpretaci. Označení formální koncept má vyjádřit, že jde o matematickou strukturu, které do jisté míry odpovídá pojem v lidské mysli. Matematickou strukturou, která je vhodným prostředkem reprezentace pojmů (konceptů) v daném kontextu, je zde *konceptový svaz*.

Autory formální teorie konceptů, která zde bude nastíněna, jsou Ganter a Wille [1].

11.2 SVAZ

Definice 11.1

Algebraickou strukturu tvoří množina s jednou nebo více na ní definovanými operacemi. *Svaz* je algebraická struktura sestávající z množiny **L** s *částečným uspořádáním* $\leq$ a dvěma binárními operátory $\cup$ a $\cap$.

Jsou-li x, y prvky **L**, $x \cap y$ se nazývá *infimem* x a y, $x \cup y$ *supremem* x a y.

Pro libovolné prvky x, y, z z **L** splňují operátory $\cup$ a $\cap$ tyto axiómy :

- $x \cap y \leq x$, $x \cap y \leq y$

- je-li z prvek z **L**, pro nějž platí z ≤ x i z ≤ y, potom z ≤ x∩y
- x ≤ x∪y, y ≤ x∪y
- je-li z prvek z **L**, pro nějž platí x ≤ z i y ≤ z, potom x∪y ≤ z

Operátory ∪ a ∩ označením odpovídají množinovému sjednocení a průniku, neboť všechny podmnožiny *univerzální množiny* U tvoří svaz s částečným uspořádáním ⊂..

Definice 11.2

Ohraničený svaz je svaz s prvky T (*top*) a ⊥ (*bottom*), kde pro všechny prvky a svazu platí ⊥≤ a ≤ T.

Všechny konečné svazy jsou ohraničené a též mnohé nekonečné. Ve svazu podmnožin univerzální množiny U je T univerzální množina U a ⊥ prázdná množina.

Pro svazové operátory platí podobně jako pro množiny tyto zákony :
Idempotence : x∩x stejně jako x∪x je identické x.
Komutativnost : x∩y je identické y∩x, x∪y je identické y∪x
Asociativnost : x∩(y∩z) je identické (x∩y)∩z, x∪(y∪z) je identické (x∪y)∪z
Absorbce : x∩(x∪y) je identické x, x∪(x∩y) je identické x.

Uvedené zákony platí pro všechny svazy. V *distributivních svazech* platí navíc *distributivní zákon*. V *komplementárních svazech* je zaveden *operátor komplementu* a platí *DeMorganovy zákony*. Svaz podmnožin nějaké množiny je příkladem ohraničeného distributivního komplementárního svazu.

Všechny *hierarchické struktury konceptů* pro účely reprezentace znalostí jsou částečná uspořádání, řada z nich jsou svazy.

11.3 FORMÁLNÍ KONCEPT V KONTEXTU

11.3.1 Formální kontext a koncept

Základním datovým formátem formální konceptové analýzy je datová tabulka nazývaná *formální kontext*, určující jakými atributy se jednotlivé dané objekty nebo podmnožiny uvažované množiny objektů vyznačují.

Podle teorie Gantera a Willeho (dále G-W) [1] je koncept definován svou intenzí a extenzí v rámci daného *kontextu* vymezeného v rámci nějakého modelovaného světa – referenčního systému.

Definice 11.3

Formální kontext je struktura určená trojicí (**O**,**A**,***I***), kde **O** je množina objektů, **A** je množina atributů a ***I*** ⊆ **O** × **A** je binární relace.

Relace ***I*** má vyjadřovat, že nějaký objekt o má vlastnost danou atributem a. Dvojice (o,a) ∈ ***I*** tedy znamená, že objekt o se vyznačuje atributem a.

Každá množina objektů **C** ⊆ **O** je spojena s podmnožinou množiny **A** atributů - *intenzí*

$$\text{int}(\mathbf{C}) = \{a \in \mathbf{A} \mid \forall o \in \mathbf{C}.(o,a) \in \boldsymbol{I}\}.$$

Analogicky každá množina atributů $\mathbf{B} \subseteq \mathbf{A}$ je spojena s množinou objektů - *extenzí*

$$\text{ext}(\mathbf{B}) = \{o \in \mathbf{O} \mid \forall a \in \mathbf{B}.(o,a) \in \boldsymbol{I}\}.$$

Stručně řečeno, intenze množiny **C** je vyjádřena množinou společných atributů, extenze množiny **B** je dána množinou objektů, které mají společné vlastnosti. To vede k pojmu formálního konceptu v G-W teorii.

Definice 11.4

Formální koncept v kontextu $(\mathbf{O},\mathbf{A},\boldsymbol{I})$ je pár (obj, attr), kde obj $\subseteq \mathbf{O}$, attr $\subseteq \mathbf{A}$ tak, že platí obj = ext(attr) a attr = int(obj).

Podle G-W tedy koncept ***C*** má svoji intenzionální část - formalizovanou množinu atributů attr a extenzionální část určenou množinou objektů obj, které se danými atributy vyznačují. Obě části určení konceptu závisí na daném kontextu.

Poznámka :

Z hlediska formální logiky lze koncept je pojímat jako trojici (C, int(C), ext(C)), kde C je jméno konceptu, int(C) je teorie spojená s konceptem C nad logikou Σ, ext(C) je model této teorie. Potom lze uvažovat třídy ekvivalence teorií jako intenze konceptů.

11.3.2 Konceptový svaz

Na množině konceptů lze definovat částečné uspořádání jako

$$(\text{obj1}, \text{attr1}) \leq (\text{obj2}, \text{attr2}) \Leftrightarrow \text{obj1} \subseteq \text{obj2} \Leftrightarrow \text{attr1} \supseteq \text{attr2}$$

Z toho je vidět, že koncepty spolu s částečným uspořádáním tvoří svaz, který má nejmenší prvek (ext(**A**), **A**) a největší prvek (**O**, int(**O**)).

Struktura $\boldsymbol{S} = (\mathbf{O},\mathbf{A},\boldsymbol{I})$ je popsána pomocí tzv. hlavní věty konceptových svazů (viz G-W).

11.4 UZÁVĚROVÉ VLASTNOSTI KONCEPTŮ

11.4.1 Uzávěry a uzávěrové prostory

Definice 11.5

Operátor φ je *uzávěrový operátor*, jestliže splňuje tři základní axiómy :

1. $X \subseteq X \,.\, \varphi$,
2. $X \subseteq Y \Rightarrow X \,.\, \varphi \subseteq Y.\, \varphi$
3. $X \,.\, \varphi \,.\, \varphi \subseteq X \,.\, \varphi$.

Definice 11.6

Množina $X \subseteq \mathbf{U}$ (podmnožina universa diskursu **U**) je *uzavřená*, je-li $X \,.\, \varphi = X$.
Dvojice $(\mathbf{U}, \varphi)$ se nazývá *uzávěrový prostor*.

Následující vlastnosti uzávěrů vyplývají přímo z axiómů 1-3 definice 11.5.

a) je-li C uzavřená a $X \subseteq C \subseteq X \,.\, \varphi$, potom $C = X \,.\, \varphi$ (tj. $X \,.\, \varphi$ je nejmenší uzavřená množina obsahující X).
b) $X \,.\, \varphi \cup Y.\, \varphi \subseteq (X \cup Y) \,.\, \varphi$
c) $(X \cap Y) \,.\, \varphi \subseteq X \,.\, \varphi \cap Y.\, \varphi$
d) $X \,.\, \varphi \cap Y.\, \varphi$ je uzavřená.
e) $\emptyset \,.\, \varphi = \emptyset$
f) $\mathbf{U} \,.\, \varphi = \mathbf{U}$

11.4.2 Uzávěry a koncepty

Definice 11.7

Nechť ***R*** je daný kontext.

Uzávěr φ_R *objektu* o̲ množiny **O** vzhledem k ***R*** je maximální množina objektů, která sdílí stejné atributy jako o̲.

Uzávěrψ_R ***atributu*** ***a̲*** množiny **A** vzhledem k ***R*** je maximální množina atributů, kterou sdílejí stejné objekty jako a̲.

Uzávěr podmnožiny $O \subseteq$ **O** *objektů* o v daném kontextu ***R*** je

$\varphi_R = \{o \mid \forall o, o \in O \Rightarrow (o,a) \in \boldsymbol{R}\}$,

podobně je *uzávěr podmnožiny* $A \subseteq$ **A** *atributů* a v daném kontextu ***R***

$\psi_R = \{a \mid \forall a, a \in A \Rightarrow (o,a) \in \boldsymbol{R}\}$

G-W ukázali, že uzávěry φ_R a ψ_R tvoří Galoisovo spojení, reprezentující koncepty, jsou isomorfní a lze je reprezentovat svazem L_R uzavřených množin, které jsou částečně uspořádány inkluzí. Návěštím každého uzlu je pár uzavřených množin, spojených Galoisovým spojením, např. v kontextu ***R*** z příkladu 11.1 následujícího odstavce jde o dvojice jako (abg, 123), reprezentující koncepty v daném kontextu ***R***. Množina abg je uzavřená v **A**, množina 123 je uzavřená v **O**.

11.5 KONCEPTOVÉ GRAFY

11.5.1 Vizuální model kontextu

Koncept je možno v daném kontextu pojímat (viz předcházející odstavec) jako základní prvek myšlení sestávající ze dvou částí – extenze a intenze. Extenze pokrývá všechny objekty náležející konceptu a intenze shromažďuje všechny atributy, kterými se koncept vyznačuje.

Pro snadné pochopení způsobu grafické reprezentace konceptů a jejich atributů, tak jak je zvykem ve formální konceptové analýze, poslouží následující řada příkladů z literatury.

Při tvorbě minimálního svazu byla v následujících příkladech aplikována metoda formální konceptové analýzy FCA (Ganter – Wille) [1], vycházející z definice 11.4 konceptu v daném kontextu.

Příklad 11.1

Příklad Johna L Pfalze a Christophera M. Taylora [3] :

Je dán kontext ***R*** (tab. 11.1), tj. relace mezi množinou atributů A (tab. 11.2 b)) a množinou objektů O (tab. 11.2 a)).

R	a	b	c	d	e	f	g	h	i
1	×	×					×		
2	×	×					×	×	
3	×	×	×				×	×	
4	×		×				×	×	×
5	×	×		×		×			
6	×	×	×	×		×			
7	×		×	×	×				
8	×		×	×		×			

tab. 11.1

Význam uvedených množin objektů **O** a atributů **A** kontextu ***R*** je dán tabulkou 11.2.

	objekty
1	pijavice
2	cejn
3	žába
4	pes
5	plevel
6	třtina
7	bob
8	kukuřice

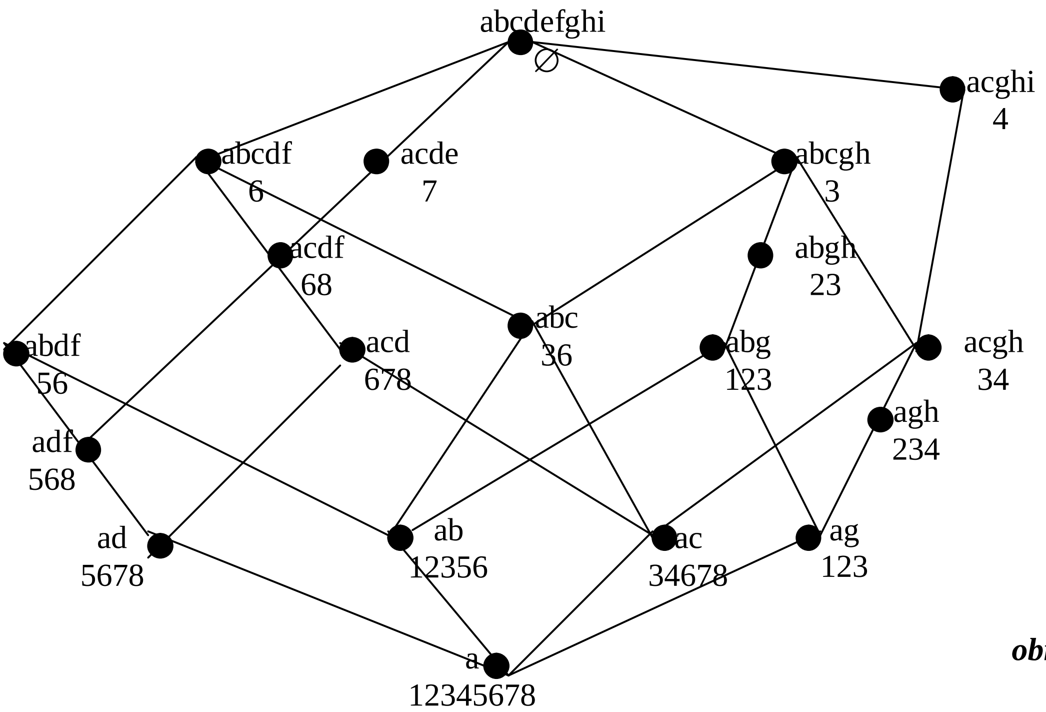
	atributy
a	potřebuje k životu vodu
b	žije ve vodě
c	žije na souši
d	potřebuje k tvorbě potravy chlorofyl
e	klíčí dvěma malými lístky
f	klíčí jedním malým lístkem
g	může se pohybovat
h	má větve
i	kojí potomky

tab. 11.2 a),b)

V obr. 11.1, zobrazujícím graf konceptového svazu L_R kontextu ***R***, je svaz orientovaný vzhledem k **A**, kde top (abcdefghi, ∅), tj. universum diskursu, které musí být uzavřené, je supremem svazu. Atribut a, který přísluší každému objektu, je infimem svazu, zde má návěští (a, 12345678).

Každý koncept začleněný do grafu představuje uzel se dvěma částmi návěští : V dolní části návěští je uvedena extenze konceptu, tj. objekty, které koncept reprezentuje, a v horní části návěští jeho intenze, tj. množina atributů, kterými se koncept vyznačuje.

Svaz L_R na obr. 11.1 je vizuálním modelem obsahu kontextu ***R***.



obr. 11.1

Příklad 11.2

Tento příklad, jehož autory jsou Bernhard Ganter a Gerd Stumme [2], ukazuje, jak lze grafickou reprezentací konceptového svazu přehledným způsobem nahradit informace z tabulky (kontextu) ***Letecké spoje*** (tab. 11.3).

Na rozdíl od obr. 11.1, v němž jsou u každého uzlu grafu uvedeny obě části návěští - horní reprezentující intenzi konceptu a dolní reprezentující jeho extenzi, zde bude ukázán mnohem úspornější způsob popisu konceptů v grafu jejich intenzemi a extenzemi.

LETECKÉ_SPOJE	Latin America	Europe	Canada	Asia Pacific	Middle East	Africa	Mexico	Caribean	United states
AIR CANADA	×	×	×	×	×		×	×	×
AIR NEW ZEALAND		×		×					×
ALL NIPPON AIRWAYS		×		×					×
ANSETT AUSTRALIA				×					
THE AUSTRALIAN AIRLINES GROUP		×	×	×	×	×			×
BRITISH MIDLAND		×							
LUFTHANSA	×	×	×	×	×	×	×		×
MEXICANA	×		×				×	×	×
SCANDINAVIAN AIRLINES	×	×		×		×			×
SINGAPORE AIRLINES		×	×	×	×	×			×
THAI AIRLINES INTERNATIONAL	×	×		×				×	×
UNITED AIRLINES	×	×	×	×			×	×	×
VARIG	×	×		×		×	×		×

tab. 11.3

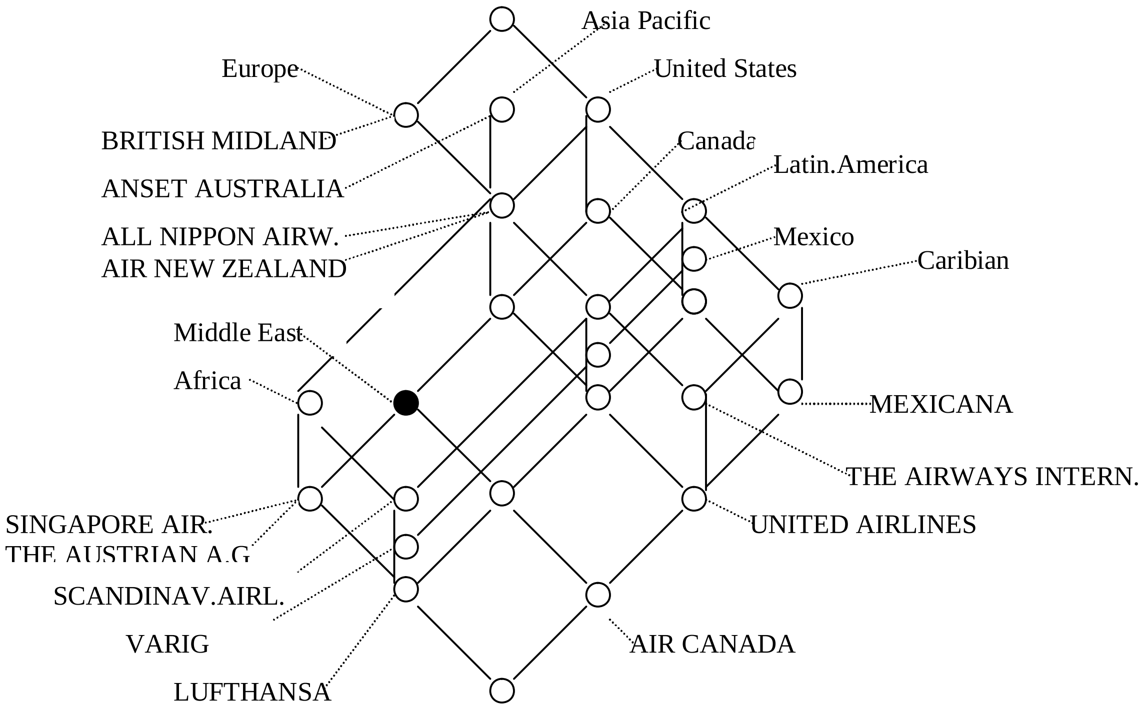
Obr. 11.2 zobrazuje konceptový svaz daného kontextu ***LETECKÉ_SPOJE***. Na rozdíl od předcházejícího příkladu v grafu na obr. 11.2 nejsou uvedeny vždy obě složky návěští konceptu příslušejícímu k danému uzlu. Chybí-li část návěští v atributové složce, lze ji najít na cestě grafen směrem vzhůru ke konceptu top, chybí-li objektová složka je třeba ji hledat na cestě grafem směrem dolů ke konceptu bottom.

V tab. 11.3, stejně jako v obr. 11.2 jsou objekty představující letecké společnosti označeny velkými písmeny, atributy představující cílová místa letů jsou označeny malými písmeny.

Pro vytváření a též čtení konceptových grafů platí pravidlo :
Objekt o má atribut a, právě když v grafu existuje cesta vzhůru od uzlu označeného o k uzlu označenému a.
Podobně atribut a náleží objektu o, právě když v grafu existuje cesta dolů od uzlu označeného a k uzlu označenému o.
Koncept C_1 je subkonceptem konceptu C_2, právě když existuje cesta grafem směrem dolů od uzlu konceptu C_2 k uzlu konceptu C_1.

Podle uvedeného pravidla je třeba tu část návěští, která se týká objektu (extenze) umístit do grafu co nejníže, neboť všechny atributy, kterými se objekt vyznačuje, lze přečíst na cestách od objektu směrem nahoru. Podobně je třeba tu část návěští, která se týká atributu, umístit v grafu co nejvýše, neboť všechny objekty, které se tímto atributem vyznačují, lze přečíst na cestách od atributu směrem dolů.

Např uzel s návěštím Middle East představuje koncept jehož extenzi {Singapore_Airlines, The_Austrian_Airlines_Group, Lufthansa, Air_Canada} lze nalézt na cestách v dolní části grafu od úrovně výplní vyznačeného uzlu. Této extenzi pak odpovídá intenze konceptu {Middle_East, Canada, United_States, Europe, Asia_pacific} představující množinu atributů, které tyto objekty sdílejí.



obr. 11.2

Příklad 11.3

E.W. Wolf [5] vysvětluje tvorbu minimálního konceptového svazu v daném kontextu (z daných dat tabulky 11.2 - ***ZIVOCICH***) na následujícím příkladě:

ZIVOCICH	**lovící**	**létající**	**pták**	**savec**
Lev	×			×
Pěnkava		×	×	
Orel	×	×	×	
Zajíc				×
Pštros			×	

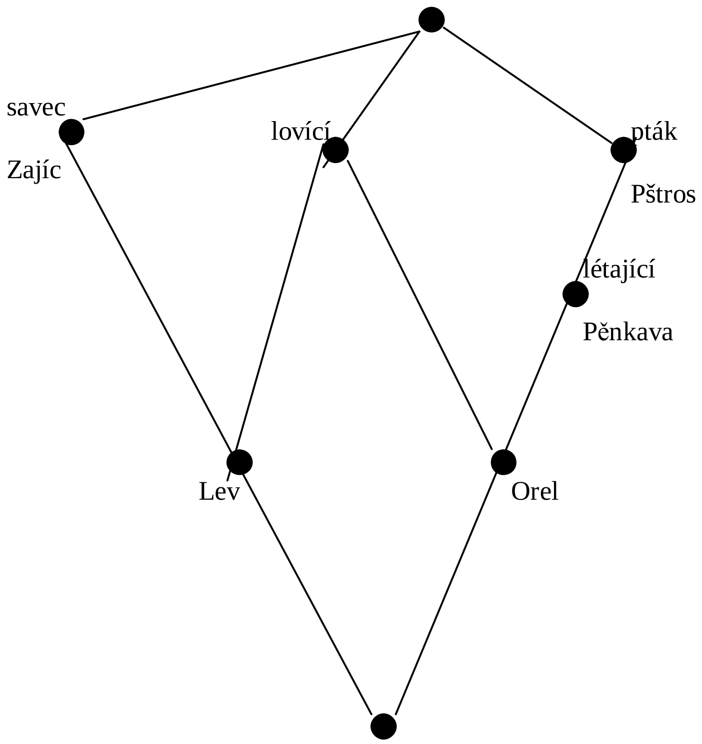
tab. 11.4

Daný kontext představuje interpretační strukturu predikátů uvedených v záhlavích tabulky ***ZIVOCICH***.

Pro množinu A = {Pěnkava, Orel}objektů je možno určit množinu B = {létající, pták} atributů, které tyto objekty sdílejí. Protože dvojici atributů {létající, pták} již nesdílí žádný další objekt v daném kontextu U, je podle definice 11.2 dvojice (A,B) *formálním konceptem* v daném kontextu.

Mezi koncepty a daným kontextem jsou zde relace subkoncept, superkoncept, např. koncept s intenzí {lovící, létající, pták}, který zde má extenzi {Orel} je subkonceptem konceptu s intenzí {létající, pták} s extenzí {Pěnkava, Orel}.

Pro každý objekt o lze sestrojit tzv. *objektový koncept* (A,B), kde B je množina atributů objektu o a A je množina objektů, pro které jsou tyto atributy charakteristické. Podobně lze sestrojit pro každý atribut a *atributový koncept* (C,D), kde C je množina objektů vyznačujících se atributem a a D je množina atributů, kterými se C vyznačuje.



obr. 11.3

Informace týkající se vzájemných souvislostí konceptů (označených zde slovy začínajícími velkými písmeny) a jejich atributů (označených zde malými písmeny) jsou graficky zobrazeny v na obr. 11.3.

11.5.2 Uspořádání vícehodnotových dat

Jak ukazují předcházející tabulky, výchozími znalostmi pro tvorbu grafu konceptuálního svazu jsou vždy tabulky kontextů. Ne každá data lze však použít pro vytvoření kontextu přímo. Následující příklad autorů [2] ukazuje transformaci tzv. vícehodnotových dat do formálního kontextu.

Příklad 11.4

Data určující kontext ***K*** jsou dána formou tabulky 11.5 :

K	pohlaví	věk
Adam	m	21
Betty	ž	50
Chris	/	66
Dora	ž	88
Eva	ž	17
Fred	m	/
George	m	90
Harry	m	50

tab. 11.5

Tabulka transformace uvedeného vícehodnotového kontextu na formální kontext :

	pohlaví		věk				
K	m	ž	<18	<40	≤65	>65	≥80
Adam	×			×	×		
Betty		×			×		
Chris						×	
Dora		×				×	×
Eva		×	×	×	×		
Fred	×						
George	×					×	×
Harry	×				×		

tab. 11.6

Pravidla transformace dvou vícehodnotových sloupců <u>pohlaví</u> a <u>věk</u> jsou patrná z těchto dvou nyní již stupnicových tabulek 11.7 a 11.8 :

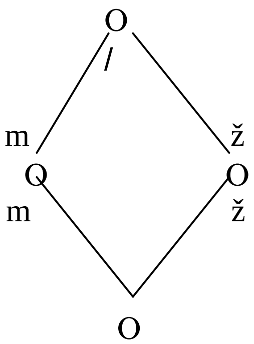
pohlaví	m	ž
m	×	
ž		×
/		

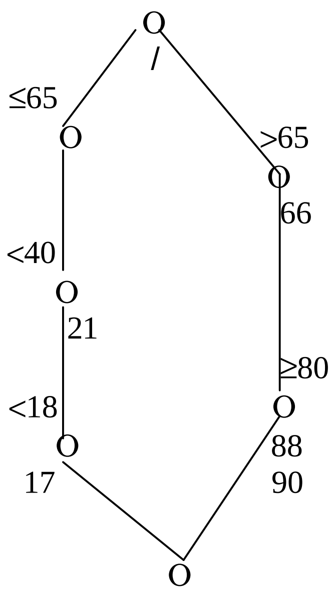
tab. 11.7

věk	<18	<40	≤65	>65	≥80
17	×	×	×		
21		×	×		
50			×		
66				×	
88				×	×
90				×	×
/					

tab. 11.8

Grafy odpovídající tabulkám kontextů ***pohlaví*** a ***věk***:

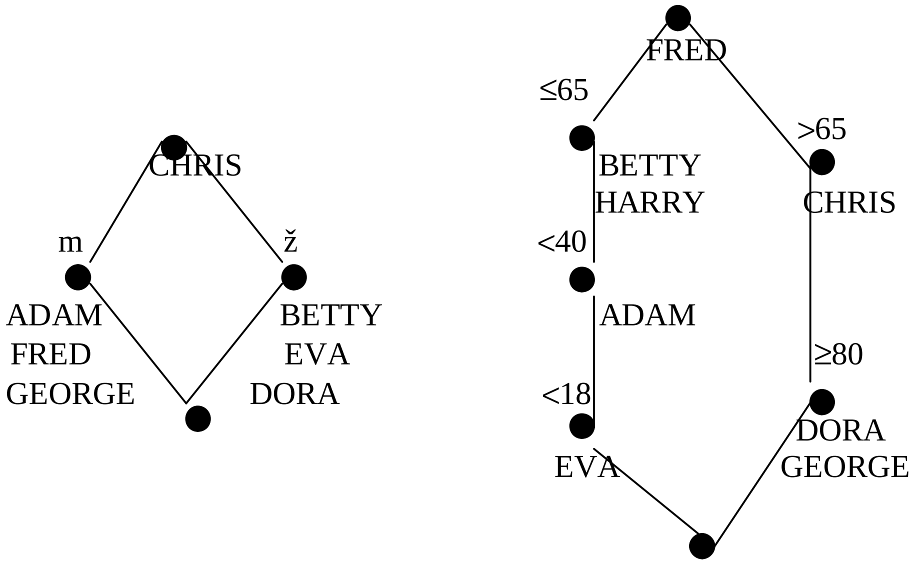




obr. 11.4 a),b)

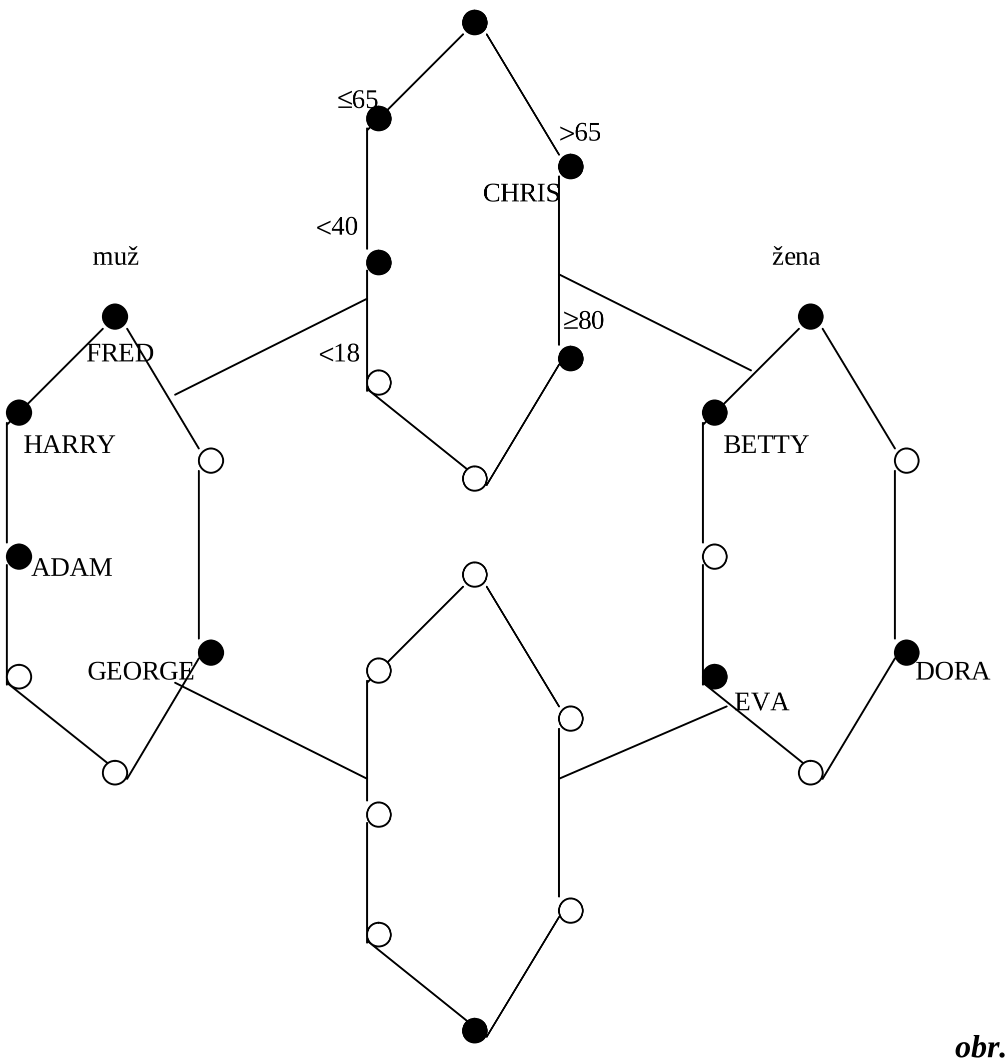
V obou grafech / nemá atributy, proto představuje top koncept. Graf stupnice věk je rozdělen na dvě větve reprezentující mladé a staré osoby. Nyní je možno transformovat grafy do stupňovaných kontextů pohlaví a věku takto :

Tyto grafy lze doplnit uvažovanými objekty (obr. 11.5) :



obr. 11.5 a),b)

Vzájemné vztahy dvou nebo více vícehodnotových atributů lze zobrazit též pomocí „zahnízdění“ grafů :

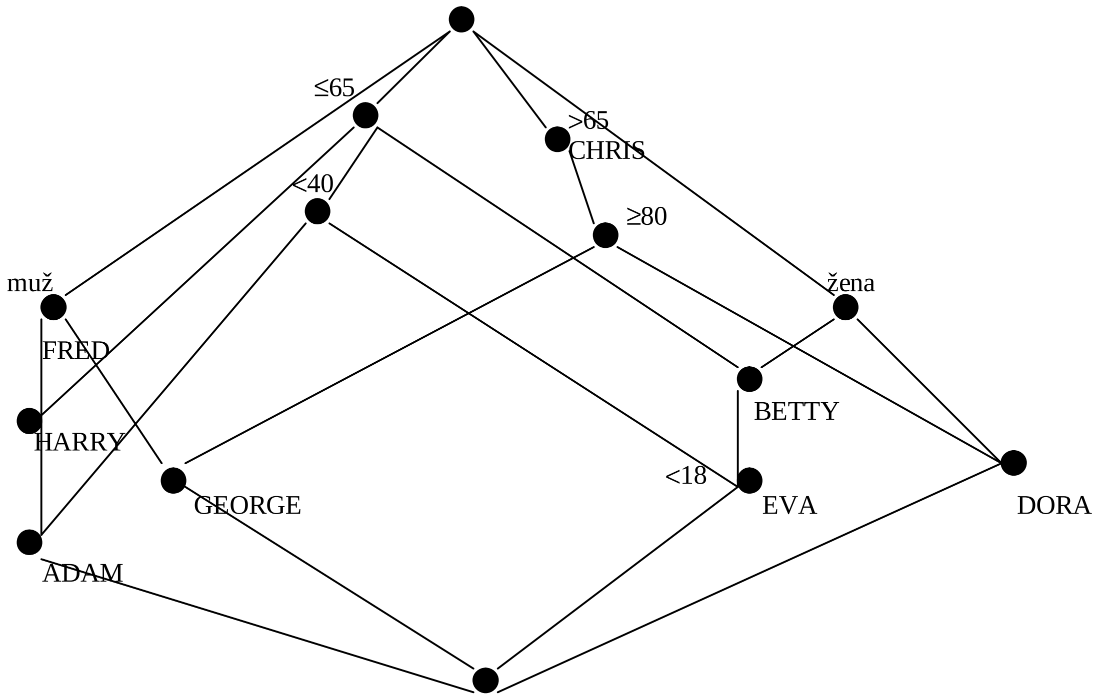


obr. 11.6

Nevyplněné kroužky, k nimž nejsou přiřazeny dané objekty, umožňují formulovat obecné implikace. Např. „každá žena mladší než 18 je mladší než 40“.

Uvedená metodika umožňuje rychlou hrubou orientaci v datech lépe než jiné statistické metody znázornění.

Graf konceptového svazu kontextu z tab. 11.6 :



obr. 11.7

Formální konceptová analýza (FCA) se provádí za účelem nalezení konceptů v daném kontextu (v datech). Matematickou strukturou, která je vhodným prostředkem reprezentace konceptů v daném kontextu, je konceptuální svaz.

Algebraickou strukturu tvoří množina s jednou nebo více na ní definovanými operacemi. *Svaz* je algebraická struktura sestávající z množiny **L** s částečným uspořádáním $\leq$ a dvěma binárními operátory $\cup$ a $\cap$. Jsou-li x, y prvky **L**, $x \cap y$ se nazývá infimem x a y, $x \cup y$ supremem x a y.

Základním datovým formátem formální konceptové analýzy je datová tabulka ***I*** nazývaná *formální kontext*, určující jakými atributy se jednotlivé dané objekty nebo podmnožiny uvažované množiny objektů vyznačují.

Formální kontext je struktura určená trojicí (**O**,**A**,***I***), kde **O** je množina objektů, **A** je množina atributů a $\boldsymbol{I} \subseteq \mathbf{O} \times \mathbf{A}$ je binární relace.

Intenze množiny **C** je vyjádřena množinou společných atributů, *extenze* množiny **B** je dána množinou objektů, které mají společné vlastnosti. To vede k pojmu *formálního konceptu* v G-W teorii.

Formální koncept v kontextu (**O**,**A**,***I***) je pár (obj, attr), kde obj $\subseteq$ **O**, attr $\subseteq$ **A** tak, že platí obj = ext(attr) a attr = int(obj).

Kontrolní otázky :

1. Co je formální kontext ?
2. Charakterizujte intenzi a extenzi konceptu.
3. Jak lze z daného kontextu poznat, že daná podmnožina objektů je extenzí formálního konceptu ?

Literatura

1. Ganter, B., Wille, R. : Formal Concept Analysis : Mathematical Foundations. Springer Verlag, Berlin, 1999.
2. Ganter, B., Stumme, G. : Formal Concept Analysis : Methods and Applications in Computer Science.
   http://www.aibf.uni-karlsruhe.de/WBSlgst/FBA03/ue05.pdf
3. Pfalz, J.L., Taylor, Ch.M. : Scientific Knowledge Discovery through Iterative Transformation of Concept Lattices.
   Proc. Workshop of Discrete Math. And Data Mining, April 2002, 65-74.
4. Wolf, E. W. : A First Course in Formal Concept Analysis. How to understand diagrams.
   In : Baulbaum, F. (ed) SoftStat’93. Advances in Statistical software 4, 429-438.

12 EMPIRICKÁ INDUKCE NA KONCEPTOVÝCH GRAFECH

Tato lekce, jejíž prostudování by vám mělo trvat zhruba 1,5 h, navazuje na lekci předcházející, s tím že metody analýzy konceptových svazů staví na teoretický základ. Dozvíte se zde též, jak postupovat při analýze daného kontextu za účelem získání vědeckých znalostí, tj. implikací mezi sledovanými atributy, resp. objekty.

12.1 DOLOVÁNÍ VĚDECKÝCH ZNALOSTÍ Z DAT

12.1.1 Tradiční pojetí dolování znalostí z dat

Pojmem *dolování znalostí z dat* se zpravidla rozumí vyhledávání podmnožin množiny sledovaných objektů, reprezentovaných zpravidla řádky relace, které jsou si navzájem podobnější než je tomu v případě objektů z navzájem různých podmnožin. Nalezený rozklad množiny objektů na podmnožiny - *shluky*, resp. hierarchické struktury shluků, pak představuje nějakou hypotézu o *klasifikační struktuře* objektů reprezentovaných daty, která je-li akceptovatelná, může představovat novou znalost o zkoumané množině objektů. Znalost klasifikační struktury dat pak umožňuje, aby nový objekt byl k některému z výsledných shluků přiřazen s tím předpokladem, že sdílí i další interpretující charakteristické vlastnosti objektů tohoto shluku.

V jiných případech je dolování znalostí z dat zaměřeno na zkoumání vzájemných závislostí atributů, charakterizujících zkoumané objekty, za účelem zjištění, zdali neexistují nějaké vzájemně závislé skypiny atributů, představující faktory, způsobující variabilitu v rámci množiny danými daty reprezentovaných objektů. Nalezené faktory zde mohou představovat novou znalost, která je dobře využitelná pro zjednodušení sledování objektů uvažovaného typu na základě menšího počtu vysoce reprezentativních atributů.

12.1.2 Dobývání znalostí z konceptových svazů

Dobývání znalostí z dat, postavené zde na algebraických a logických principech analýzy pojmů (konceptů) a jejich vzájemných vztahů, speciálně na konceptových svazech a deskripční logice, se pokouší sáhnout k samotným principům kategorizace objektů lidského myšlení, tj. pojmů (konceptů).

Paradigma těchto *vědeckých znalostí* lze charakterizovat následujícím symbolickým zápisem :

$$(\forall o \in O)((P(o) \rightarrow Q(o)) ,$$

kde O je zájmové universum diskursu, sestávající z objektů, P a Q jsou unární predikáty, které jsou konjunkcemi /disjunkcemi atributů objektů.

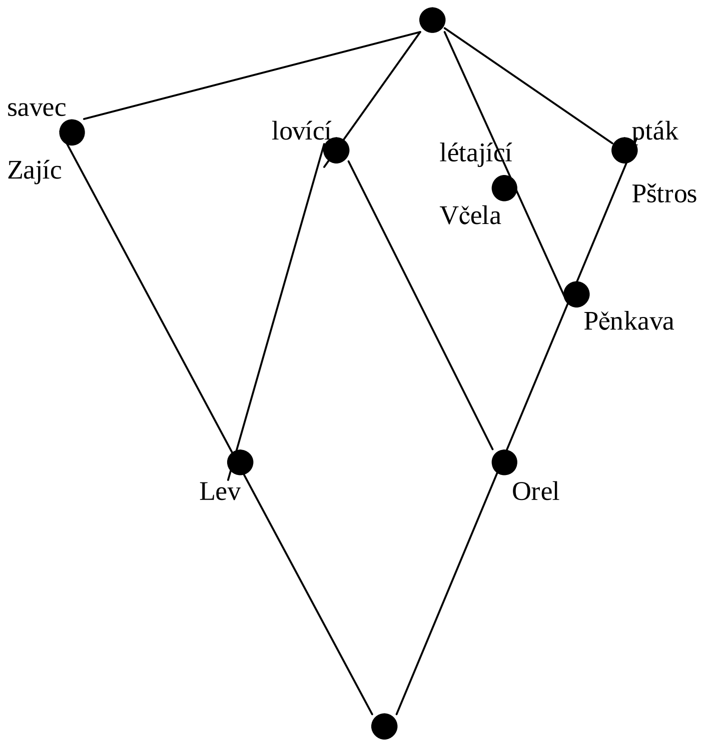
12.2 VYHLEDÁVÁNÍ IMPLIKACÍ V DANÉM KONTEXTU

12.2.1 Konceptové učení

E.W. Wolf [8], autor příkladu 11.3 uvedeného v předcházející lekci, uvádí též následující příklad "konceptového učení", který je názorným příkladem vyhledávání implikací v rámci daného kontextu na základě konceptů – uzavřených množin objektů.

Příklad 12.1

Příklad konceptového učení : Syn se učí od otce, jehož znalosti tvoří kontext ***U***, zatímco znalosti syna jsou pouze částí ***K*** universa ***U***.



obr. 12.1

Objekty a atributy kontextu ***K*** = ***ZIVOCICH*** jsou zároveň objekty a atributy ***U*** – syn vychází z kontextu ***K*** a uvědomí si, že v tomto kontextu je každé létající zvíře ptákem, tj. platí

$$\text{létající}(x) \rightarrow \text{pták}(x).$$

Táže se otce, zdali tato implikace platí i v **U**. Otec odpoví záporně a uvede protipříklad, např. včelu, která má v daném kontextu pouze atribut létá.

Zdá se, že v tomto novém kontextu platí další implikace „lovení a pták implikuje létání“, tj.

$$\text{lovící}(x) \sqcap \text{pták}(x) \rightarrow \text{létající}(x).$$

Syn se táže otce, zdali to platí v universu ***U***. Odpověď je kladná. Ale chytrý syn pozná, že v tomto kontextu též platí implikace „savec a létání implikuje lovení a pták“, tj.

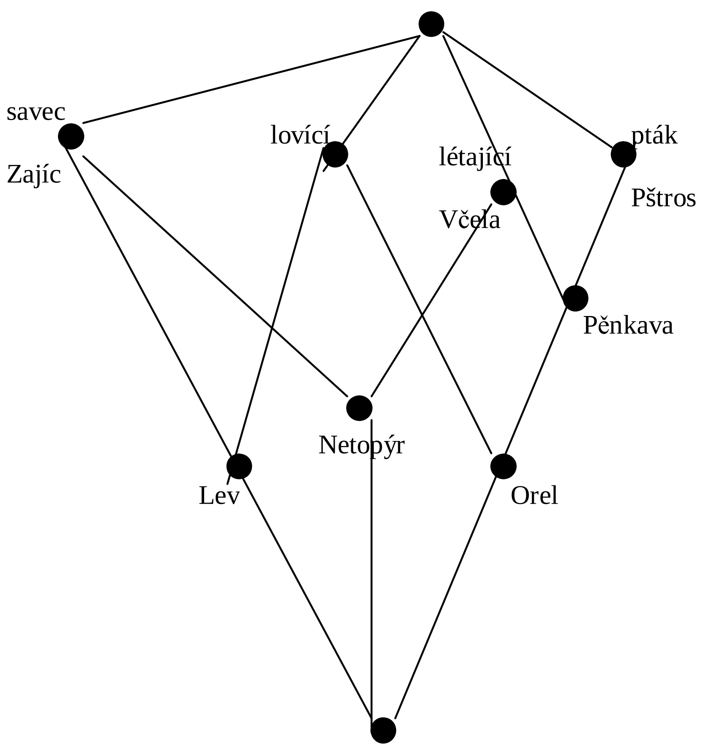
$$\text{savec}(x) \sqcap \text{létající}(x) \rightarrow \text{lovící}(x) \sqcap \text{pták}(x).$$

Platí to i v universu ***U***? Otec odpoví ne, a jako protipříklad uvede netopýra, který létá, ale neloví.

Nové znalosti vycházející nyní z kontextu ***ZIVOCICH2*** (tab. 12.1) nyní reprezentuje graf na obr. 12.2.

ZIVOCICH2	**lovit**	**létat**	**pták**	**savec**
Lev	×			×
Pěnkava		×	×	
Orel	×	×	×	
Zajíc				×
Pštros			×	
Včela		×		
Netopýr		×		×

tab. 12.1



obr. 12.2

Otec vidí, že konceptuální svaz jeho syna má tutéž strukturu jako jeho – jsou vzájemně isomorfní. Konceptuální svaz ***U*** má mnohem více objektů, ale všechny lze aranžovat do grafu na obr. 12.2.

12.2.2 Generátor konceptů

Téměř intuitivní pojetí vyhledánání implikací z konceptových svazů, tak jak bylo předvedeno na předcházejícím příkladě, je třeba postavit na teoretičtější základ, kterým zde bude pojem generátoru konceptů.

Hlavní ideou uzávěrových prostorů je *generátor uzavřené množiny*.

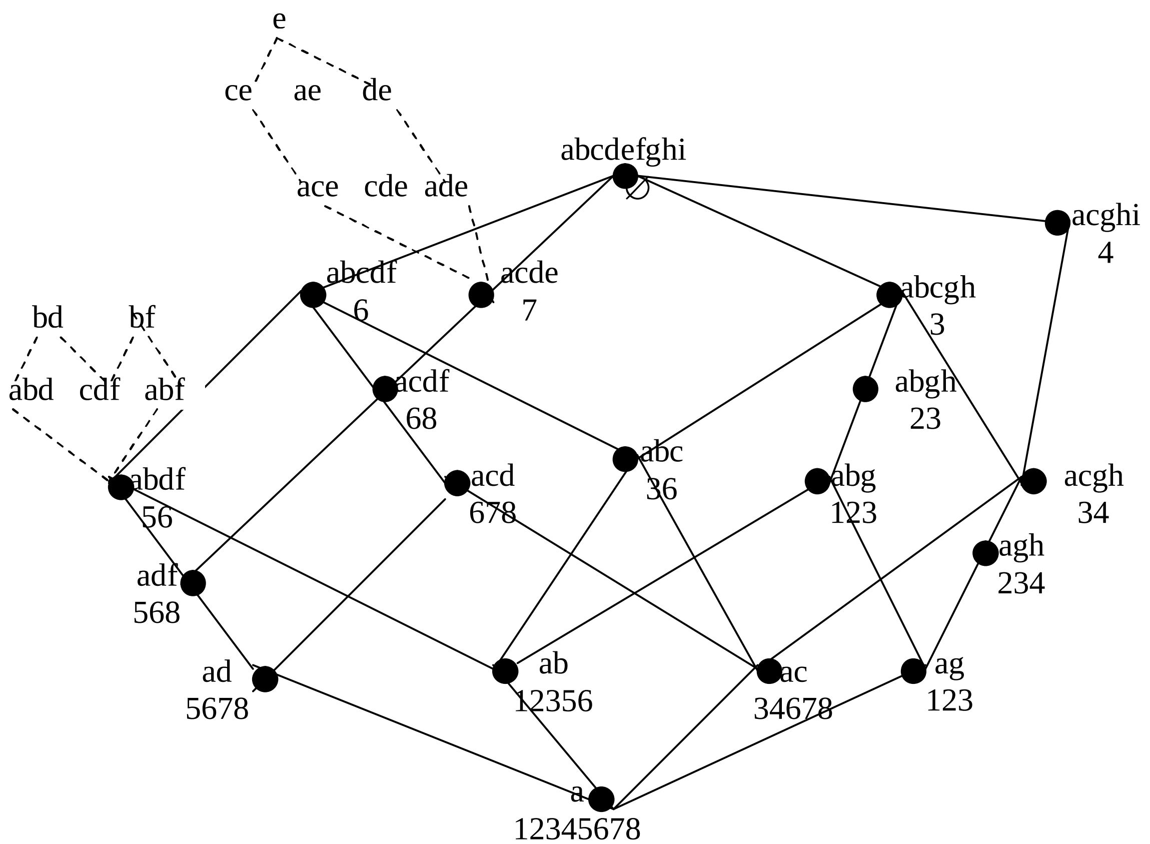
Generátor uzavřené množiny Z je minimální (vzhledem k uspořádání inkluzí) množina X taková, že X **.** φ = Z.

Z **.** γ_i je i-tý generátor Z, Z **.** Γ je soubor { Z **.** γ_i } všech generujících množin.

S využitím vlastností konceptů z uzávěrového prostoru lze generovat konceptový svaz a současně určovat generátory těchto uzavřených konceptů.

Např. v konceptovém svazu L_R kontextu ***R*** na obr. 11.1 jednotlivý atribut e generuje uzavřený koncept acde, tj. {e}**.** ψ_R= {acde}. V rámci ***R*** to znamená, že každý objekt, který má vlastnost e (zde jediný), má též vlastnosti a,c,d. podobně lze zjistit, že {bd}, stejně jako{bf} generuje {abdf}. V daném příkladě atributy bh generují abgh, neboť atributy bh se vyznačují objekty 23, které navíc sdílejí i atributy ag. V tomto kontextu tedy lze zapsat bh $\Rightarrow$ abgh. Podobně lze ukázat, že bcd a bcf jsou minimální generátory abcdf, tj. bcd $\vee$ bcf $\Rightarrow$ abcdf. To dává možnost nového způsobu dobývání znalostí z relací (kontextů).

Obr. 12.3 byl vytvořen z obr. 11.1 zakreslením (čárkovanou čarou) výše uvedených generátorů uzavřených množin atributů.



obr. 12.3

12.2.3 Induktivní transformace

Jde o transformaci konceptového svazu po přidání nového objektu se svými atributy do relace představující daný kontext. To vede k zajímavým vztahům mezi uzavřenými množinami Z = X **.** φ a jejich generátory Z **.** γ.

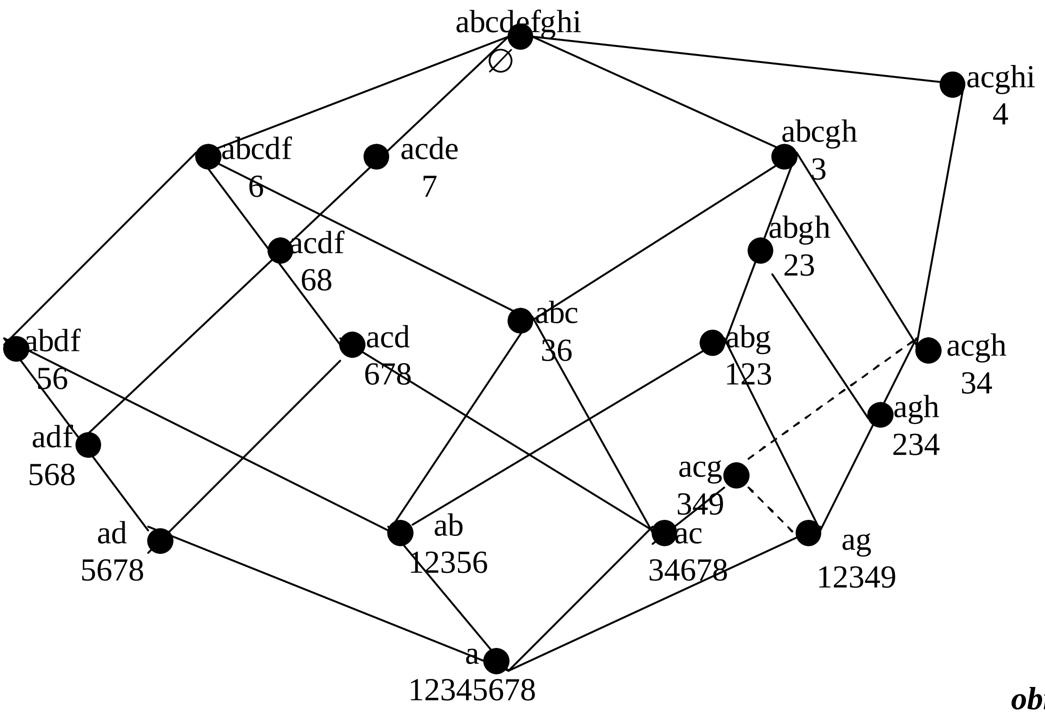
Každé přidání řádku relace do ***R*** přidá nejméně jednu uzavřenou množinu do L_R, neboť atributy jednotlivého řádku tvoří uzavřenou množinu v **A**. Označme tento řádek (o´, A´). Existuje-li $Z \in L_R$ takové, že A´= Z, pak svaz zůstává nezměněn. V opačném případě existuje alespoň jedna uzavřená Z v L_R taková, že A´$\subset$ Z. Uvažujme A´$\cap$ Y pro všechny Y, Y$\subset$ Z. Tyto jsou pouze prvky konceptového svazu L_R, s nimiž může A´ být v interakci.

Např. přidáním do kontextu ***R*** (tab. 11.1) z příkladu 11.1 nového řádku 9 s atributy acg vede k vytvoření nového konceptového svazu L_{R2}, odpovídajícího kontextu R2 (tab.12.2).

R2	a	b	c	d	e	f	g	h	i
1	×	×					×		
2	×	×					×	×	
3	×	×	×				×	×	
4	×		×				×	×	×
5	×	×		×		×			
6	×	×	×	×		×			
7	×		×	×	×				
8	×		×	×		×			
9	×		×				×		

tab. 12.2

Je zřejmé, že acg $\subset$ acgh a acg $\cap$ agh = ag. Tento přírůstek v datech změnil generační strukturu L_R. V L_R máme (cg $\vee$ ch) $\Rightarrow$ acgh. V L_{R2} nyní máme cg $\Rightarrow$ acg, tedy cg už nemůže být generátorem acgh. Nyní platí pouze ch $\Rightarrow$ acgh.



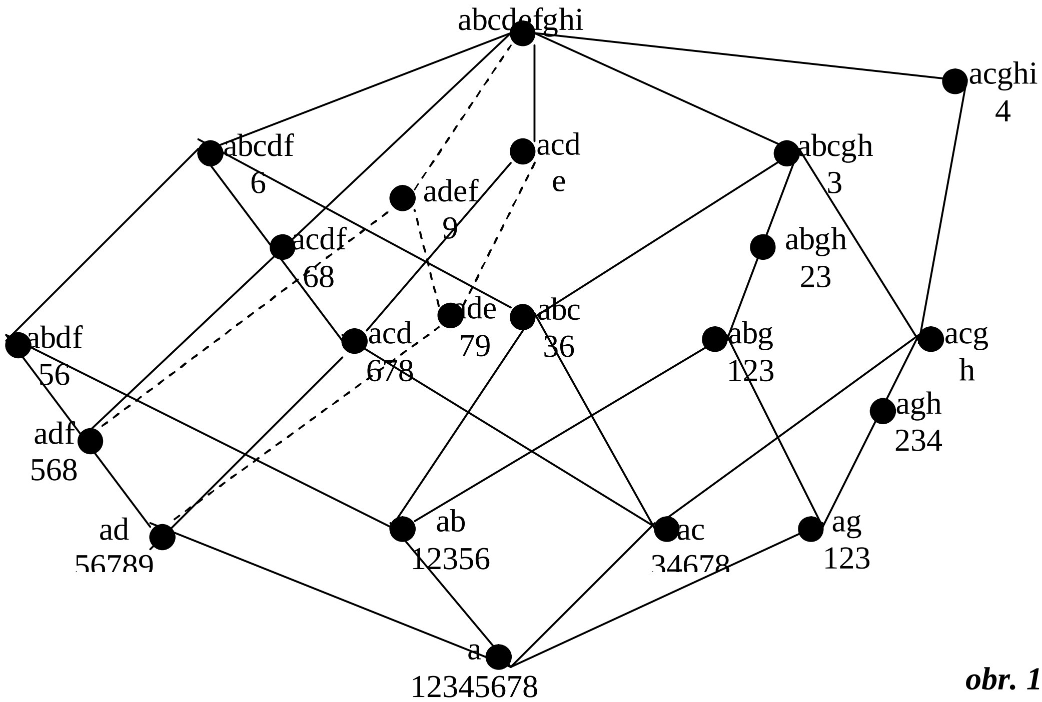
obr. 12.4

Je vidět (obr. 12.4), že tento nový objekt není příliš rozdílný od existujících objektů. Je obsažen v Z = acgh, který je dosti nízko v L_R. Předpokládejme Z = abcdefghi = U, tj. universu atributů. Lze ukázat, že ef je generátorem abcdefghi spolu s 11 dalšími minimálními generátory. Ale zde nemáme žádný objekt ve spojitosti s abcdefghi = A. V tomto kontextu ef je logickou kontradikcí.

V tab. 12.3 je objekt 9 změněn tak, že má atributy adef. Kombinace ef již nepředstavuje kontradikci. V L_{R3} (obr. 12.5) adef je pokryto Z = U a protíná acde a abcdf (které jsou rovněž pokryty Z) v ade, resp. adf. Uzavřená množina ade je nová a rekursivně protíná acd (pokryté acde) jako ad.

R3	a	b	c	d	e	f	g	h	i
1	×	×					×		
2	×	×					×	×	
3	×	×	×				×	×	
4	×		×				×	×	×
5	×	×		×		×			
6	×	×	×	×		×			
7	×		×	×	×				
8	×		×	×		×			
9	×			×	×	×			

tab. 12.3

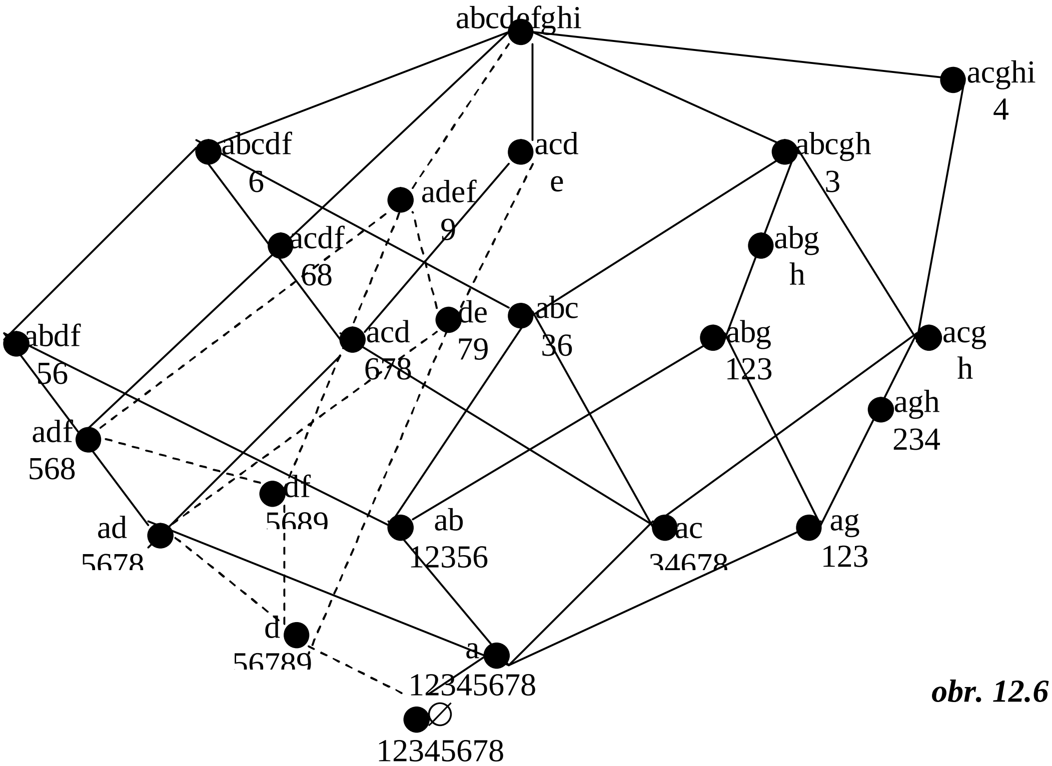


obr. 12.5

R2'	a	b	c	d	e	f	g	h	i
1	×	×					×		
2	×	×					×	×	
3	×	×	×				×	×	
4	×		×				×	×	×
5	×	×		×		×			
6	×	×	×	×		×			
7	×		×	×	×				
8	×		×	×		×			
9				×	×	×			

tab. 12.4

Předp. dále, že <u>a</u> je atributem všech objektů. To odpovídá logické tautologii v universu R. Zavedením objektu 9 s atributy pouze def (tab 12.4) se to změní, neboť dochází k průniku s acde a abcdf, čímž se vytvoří de a df a konceptový svaz se zajímavě změní (obr 12.6).



obr. 12.6

Tradičně se pojmem *dolování znalostí z dat* se zpravidla rozumí vyhledávání *shluků*, tj. podmnožin vzájemně si podobných objektů, nebo zkoumání vzájemných závislostí atributů, charakterizujících zkoumané objekty, za účelem zjištění, zdali existují nějaké vzájemně závislé skypiny atributů, představující *faktory* variability dat.

Paradigma *vědeckých znalostí* lze charakterizovat následujícím symbolickým zápisem:

$$(\forall o \in O)((P(o) \rightarrow Q(o)) ,$$

kde O je zájmové universum diskursu, sestávající z objektů, P a Q jsou unární predikáty, které jsou konjunkcemi /disjunkcemi atributů objektů.

Dobývání znalostí z dat, postavené na algebraických a logických principech analýzy pojmů (konceptů) a jejich vzájemných vztahů, speciálně na konceptových svazech a deskripční logice, se pokouší sáhnout k samotným principům kategorizace objektů lidského myšlení, tj. pojmů (konceptů).

Pomocí generátorů uzavřených množin lze získat znalosti o implikacích mezi skupinami atributů, resp. objektů

Kontrolní otázky

4. Co se rozumí vědeckou znalostí ?
5. Co jsou generátory uzavřených množin ?

Literatura

6. Bělohlávek, R., Sklenář, V., Zacpal, J. : Concept lattices constrained by attribute Dependencies. Proc. Dateso, 2004, Ostrava, 2004.
7. Ganter, B., Wille, R. : Formal Concept Analysis : Mathematical Foundations. Springer Verlag, Berlin, 1999, 2004 (?).
8. Pfaltz, J.L., Jamison, R.E. : Closure Systems and their Structures. RelMiCS 2000, Valcartier, Quobec, Jan. 2000, 121-123.
9. Pfaltz, J.L. : Transformations of Concept Graphs : An Approach to Empirical Induction. Proc. 2nd Intern'l workshop of Graph Transformation and Visual Modeling Techniques, CTVM 2001, Crete, Grece, July 2001.
10. Pfaltz, J.L., Taylor, Ch.M. : Scientific Knowledge Discovery through Iterative Transformation of Concept Lattices. Workshop of Discrete math. And Data Mining, Univ. Of Virginia, April 2002, 65-74.
11. Wolf, E. W. : A First Course in Formal Concept Analysis. How to understand diagrams.
    In : Baulbaum, F. (ed) SoftStat'93. Advances in Statistical software 4, 429-438.

Obsah