Kapitola 5

Predikátová logika

Jedným z nedostatkov výrokového počtu je obtiažna reprezentácia komplexnejších konceptov. Napríklad pri vetách typu "ak v Poprade prší, potom v Košiciach svieti slnko" je potrebné zohľadňovať miesta a stavy (resp. objekty, ich vlastnosti a vzťahy medzi nimi) – a teda každá kombinácia možného miesta a možného stavu musí byť reprezentovaná jedným symbolom, čo vedie jednak k neprehľadnosti a jednak k veľkému počtu použitých symbolov.

Naproti tomu *predikátový počet* umožňuje prirodzeným spôsobom reprezentovať vlastnosti objektov a vzťahy medzi objektmi. Reprezentácia uvedenej vety by mohla mať pri použití predikátového počtu tvar

PREDIKÁTOVÝ POČET

- $dazd(poprad) \rightarrow slnko(kosice)$
- $poprad(dazd) \rightarrow kosice(slnko)$
- $pocasie(poprad, dazd) \rightarrow pocasie(kosice, slnko)$

podľa toho, či použité predikáty reprezentujú mestá a ich vlastnosti (napr. $dazd(poprad)$ – dážď je vlastnosť Popradu) alebo stavy a ich vlastnosti (napr. $poprad(dazd)$ – miesto výskytu sa stáva vlastnosťou dažďa) alebo vzťahy medzi mestami a stavmi (napr. $pocasie(poprad, dazd)$ – vzťah medzi miestom výskytu a stavom má svoje vlastné pomenovanie).

Existuje niekoľko navzájom sa líšiacich variantov predikátovej logiky. Najjednoduchší z nich je označovaný ako *predikátová logika nultého rádu.* Syntaktické pravidlá, ktoré určujú čo môžu výrazy predikátového počtu v tomto prípade obsahovať a ako môžu byť kombinované, sú dané BNF (Backus-Naurova Forma) gramatikou v Tab. 5.1.

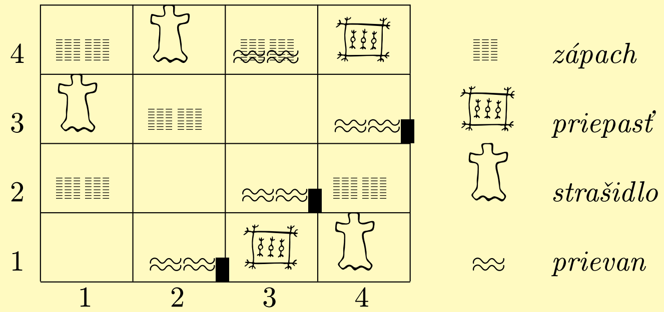
PREDIKÁTOVÝ POČET 0. RÁDU

Úlohu symbolov z výrokovej logiky preberajú teraz predikáty – to sú syntakticky najmenšie stavebné prvky, pre ktoré sa určuje ich pravdivosť

PREDIKÁTY

ILUSTRAČNÝ PRÍKLAD

Úloha: *Kamaráti Fero, Ondro, Stano, Rudo a Jaro si urobili výlet do zábavného parku. Jednou z tamojších atrakcií bol aj tajomný svet s priepasťami a strašidlami podľa nasledovného obrázka.*



Tento svet sa riadi podľa nasledujúcich zákonitostí:

1. *strašidlá a samozrejme aj priepasti nemenia svoju polohu,*
2. *ak je na nejakom poli strašidlo, tak na susedných poliach je cítiť zápach,*
3. *ak je na nejakom poli priepasť, tak na susedným poliach je cítiť prievan,*
4. *ak niekto vstúpi na pole s priepasťou, tak zahynie,*
5. *ak niekto vstúpi na pole so strašidlom, tak bude zamordovaný.*

Do miestnosti (1,1) tohto sveta vstúpili chlapci (vybavení svojimi zmyslami, pamäťou a schopnosťou logicky uvažovať) a mohli sa v ňom pohybovať vertikálnym aj horizontálnym smerom. Ich úlohou je zmapovať tento svet.

Reprezentácia: *Uvedený problém je reprezentovaný ako logická veta:*

$$\begin{array}{ll} & \forall\, S\ (prievan(S) \leftrightarrow \exists\, R\ (susedne(S,R) \wedge priepast(R))) \\ \wedge & \forall\, S\ (zapach(S) \leftrightarrow \exists\, R\ (susedne(S,R) \wedge strasidlo(R))) \\ \wedge & \forall\, A,B,C\ (susedne(pole(A,B), pole(A,C)) \leftrightarrow\ po(B,C) \vee po(C,B)) \\ \wedge & \forall\, A,B,C\ (susedne(pole(A,B), pole(C,B)) \leftrightarrow\ po(A,C) \vee po(C,A)) \\ \wedge & po(1,2) \wedge po(2,3) \wedge po(3,4) \end{array}$$

Riešenie: *Chlapci si pri návšteve nejakého poľa do pamäti uložia svoje vnemy, napr. pre pole (1,1) to bude* $\neg prievan(pole(1,1))$ *a* $\neg zapach(pole(1,1))$*. Následne sa pokúsia pre jednotlivé polia dokázať, že sú bezpečné, čo pri pobyte na (1,1) bude* $\neg priepast(pole(2,1))$, $\neg priepast(pole(1,2))$, $\neg strasidlo(pole(2,1))$ *a* $\neg strasidlo(pole(1,2))$*. Ak niektoré pole, na ktorom ešte neboli, je bezpečné, potom prejdú naň. Opakovaním tohto postupu si vytvoria nasledujúcu mapu skúmaného sveta:*

bezpečné polia:	*(1,1), (2,1), (1,2), (2,2), (2,3), (3,2), (3,3), (3,4), (4,3)*		
priepasť:	*(3,1), (4,4)*	*možná priepasť:*	*(4,2)*
strašidlo:	*(1,3), (2,4)*	*neskúmané polia:*	*(1,4), (4,1)*

Ilustr. 5.1: Príklad pre reprezentáciu predikátovou logikou

<table>
<tr><td>< veta ></td><td>::=</td><td>< predikát > | < zložená veta ></td></tr>
<tr><td>< predikát ></td><td>::=</td><td>< predikátový symbol >(< term >*)</td></tr>
<tr><td>< zložená veta ></td><td>::=</td><td>< unárny operátor > < veta ></td></tr>
<tr><td></td><td>|</td><td>< veta > < binárny operátor > < veta ></td></tr>
<tr><td></td><td>|</td><td>(< veta >)</td></tr>
<tr><td>< term ></td><td>::=</td><td>< funkcia > | < konštanta ></td></tr>
<tr><td>< funkcia ></td><td>::=</td><td>< funkčný symbol >(< term >+)</td></tr>
<tr><td>< unárny operátor ></td><td>::=</td><td>$\neg \mid \ldots$</td></tr>
<tr><td>< binárny operátor ></td><td>::=</td><td>$\vee \mid \wedge \mid \rightarrow \mid \leftrightarrow \mid \oplus \mid \uparrow \mid \downarrow \mid \ldots$</td></tr>
</table>

Tab. 5.1: Syntax predikátového počtu nultého rádu

alebo nepravdivosť. Predikát má, na rozdiel od výrokového symbolu, svoju štruktúru – má názov (reprezentovaný predikátovým symbolom) a určitý počet argumentov (označovaný ako *árnosť* predikátu). V už uvedenom zápise predikátu $poprad(dazd)$ je $poprad$ predikátovým symbolom a term $dazd$ je zase argumentom daného unárneho predikátu (árnosť rovná 1). V zápise predikátu $pocasie(poprad, dazd)$ je zrejmé, že sa jedná o binárny predikát (predikát s árnosťou 2) s predikátovým symbolom $pocasie$ a dvomi termami $poprad$ a $dazd$ vo funkcii argumentov. Ak árnosť predikátu je nulová a teda predikát nemá žiadne argumenty, potom je ekvivalentným s výrokovým symbolom.

KONŠTANTY A FUNKCIE

Argumentom predikátu môže byť iba term, ktorým je buď konštanta alebo funkcia. Konštanta sa používa pre reprezentáciu objektu bez zohľadnenia jeho štruktúry. Funkcia, podobne ako predikát, pozostáva z funkčného symbolu a argumentov, ktorých počet udáva árnosť funkcie. Oproti konštante funkcia umožňuje vyjadriť štruktúru objektov alebo objekty, ktoré nie je potrebné reprezentovať samostatným spôsobom ale je možné ich odvodiť z iných objektov. Príkladom konštanty je *dazd*, príkladom funkcie zase *letisko(poprad)*. Pri tvorbe funkcií je možné aj viacnásobné vnorenie rôznych funkcií alebo aj rekurzia tej istej funkcie, napríklad *otec(otec(jano))*. Konštantu je možné považovať za funkciu s nulovou árnosťou – teda funkciu bez argumentov.

Syntakticky zápisy funkcie a predikátu vyzerajú rovnako. Rozdiel je v tom, čo daný zápis vyjadruje. Zápis $letisko(poprad)$ je

- funkciou, ak reprezentuje objekt (popradské letisko),
- predikátom, ak sa dá rozhodnúť o jeho pravdivosti (či v Poprade je letisko alebo nie).

Výroková logika	*Predikátová logika*	*Doména*
symboly	konštanty	objekty
	funkčné symboly	funkcie
	predikátové symboly	relácie

Tab. 5.2: Základné elementy

Príklad 5.1 Pre ilustráciu uvažujme doménu celých čísel, kde tieto čísla hrajú úlohu objektov. Potom je možné objekty definovať

- priamo pomocou konštánt ($\ldots$, k_{-2}, k_{-1}, k_0, k_1, k_2, $\ldots$) alebo
- nepriamo pomocou funkcií (s funkčnými symbolmi f^+, f^-, f^*).

Unárne predikáty by napríklad mohli používať predikátové symboly *parny*, *nezaporny*, *kladny* a podobne, zatiaľ čo predikátové symboly binárnych predikátov by mohli byť $p_<$, $p_\leq$, $p_=$, $p_\geq$ a $p_>$.

Príkladom logických viet v takto definovanej predikátovej logike by mohlo byť

$$p_\geq(f^*(k_7, k_5), k_{33}) \quad \rightarrow \quad p_>(f^+(k_2, f^*(k_7, k_5)), f^+(k_1, k_{33}))$$

alebo

$$parny(k_{33}) \quad \rightarrow \quad \neg\, parny(f^+(k_1, k_{33}))$$

•

INTERPRETÁCIA KONŠTANTY

Na rozdiel od výrokovej logiky, kde interpretácia bola daná pravdivostnými hodnotami symbolov, je teraz potrebné interpretovať konštanty a funkčné ako aj predikátové symboly. Táto interpretácia sa realizuje voči nejakej doméne, ktorá obsahuje objekty, relácie a funkcie (Tab. 5.2). Interpretácia konštanty v predikátovej logike stanovuje, ktorý *objekt* domény (oblasti reálneho sveta) je reprezentovaný danou konštantou. Ak doména obsahuje m_o objektov a existuje n_k konštánt, potom je možných $(m_o)^{n_k}$ interpretácií konštánt. Pritom nie každý objekt domény musí byť reprezentovaný konštantou. Na druhej strane nejaký objekt môže byť reprezentovaný viacerými konštantami. Príkladom je skupina osôb (objekty) ktoré sú známe pod svojimi menami (konštanty). Pritom je možné, že meno niektorej osoby nikto nepoužíva alebo naopak niekto je známy pod viacerými označeniami (nielen meno ale aj prezývky).

Interpretácia funkčného symbolu znamená určiť, ktorú *funkciu* domény tento symbol popisuje, pričom pod funkciou domény sa chápe zobrazenie z $\mathcal{D}^n$ do $\mathcal{D}$, kde $\mathcal{D}$ označuje objekty domény a n árnosť funkcie. Interpretácia funkcie je teda daná interpretáciou termov (reprezentovaných funkciami alebo konštantami) ktoré sú jej argumentami, interpretáciou funkčného symbolu a celá funkcia referuje na nejaký objekt z domény. To sa dá vyjadriť ako

INTERPRETÁCIA FUNKČNÉHO SYMBOLU

$$f(t_1, ..., t_n)^I \; = f^I(t_1^I, ..., t_n^I)$$

Podobne interpretácia predikátového symbolu určuje, ktorú *reláciu* domény daný symbol popisuje, pričom pod reláciou domény sa chápe zobrazenie z $\mathcal{D}^n$ do množiny $\{TRUE, FALSE\}$. Interpretácia predikátu je daná interpretáciou termov (konštanty alebo funkcie), interpretáciou predikátového symbolu a celému predikátu je priradená pravdivostná hodnota. To sa dá vyjadriť ako

INTERPRETÁCIA PREDIKÁTOVÉHO SYMBOLU

$$p(t_1, ..., t_n)^I \; = p^I(t_1^I, ..., t_n^I) \; = TRUE/FALSE$$

Predikát je pravdivý v nejakej interpretácii I vtedy, ak tá relácia domény, na ktorú odkazuje predikátový symbol v I, platí medzi tými objektami domény, na ktoré odkazujú argumenty predikátu v danej interpretácii I.

V prípade, že árnosť predikátu je nulová a teda predikát je symbolom, potom je interpretovaný rovnako ako symbol vo výrokovej logike – interpretácia mu priamo priradí jednu z pravdivostných hodnôt.

Pre ľubovoľnú vetu F a ľubovoľnú interpretáciu I, pravdivostná hodnota F^I, ktorá je priradená vete F interpretáciou I sa určí rekurzívnym spôsobom:

INTERPRETÁCIA VETY

- ak F je konštanta, potom F^I je dané interpretáciou I,
- ak F je funkcia, potom F^I je dané interpretáciou I,
- ak F je predikát, potom F^I je dané interpretáciou I,
- ak $F = (\odot G)$ a $\odot$ reprezentuje jeden z definovaných unárnych operátorov, tak $F^I = \odot(G^I)$,
- ak $F = (G \odot H)$ a $\odot$ reprezentuje jeden z definovaných binárnych operátorov, tak $F^I = (G^I) \odot (H^I)$.

Pravdivosť vety je teda daná pravdivosťou predikátov a predpismi pre odvádzanie výslednej pravdivostnej hodnoty, platnými pre jednotlivé operátory (podľa pravdivostných tabuliek v Tab. 1.2).

Príklad 5.2 Uvažujme predikátovú logiku definovanú v predchádzajúcom príklade. Úlohou je určiť pravdivosť predikátu $parny(f^+(k_1, k_{33}))$.

Je zrejmé, že existuje nekonečne veľa interpretácií konštanty k_1 (môže reprezentovať ľubovoľné celé číslo) a rovnako aj konštanty k_{33}. Funkčný symbol f^+ može reprezentovať ľubovoľnú binárnu operáciu nad celými číslami. Predikátový symbol *parny* zase môže popisovať ľubovoľnú vlastnosť celých čísel. V tomto zmysle pri interpretácii $I = \{k_1^I{=}1,\ k_{33}^I{=}33,\ (f^+)^I{=}+$ a $parny^I{=}$párnosť$\}$ by po dosadení bolo:

$$\begin{aligned} parny(f^+(k_1, k_{33}))^I &= parny^I((f^+)^I(k_1^I, k_{33}^I)) \\ &= parnost(1+33) \\ &= parnost(34) \\ &= TRUE \end{aligned}$$

zatiaľ čo pri ľubovoľnej z interpretácií

$$I = \{k_1^I{=}3,\ k_{33}^I{=}2,\ (f^+)^I{=}+ \text{ a } parny^I{=}\text{párnosť}\}$$
$$I = \{k_1^I{=}1,\ k_{33}^I{=}33,\ (f^+)^I{=}* \text{ a } parny^I{=}\text{párnosť}\}$$
$$I = \{k_1^I{=}1,\ k_{33}^I{=}33,\ (f^+)^I{=}+ \text{ a } parny^I{=}\text{zápornosť}\}$$

by po dosadení bol daný predikát interpretovaný $FALSE$. •

Počet interpretácií

Jednotlivé interpretácie sa teda líšia v tom, ako interpretujú konštanty a symboly voči prvkom domény. Ak veta obsahuje n_k konštánt, n_f^i funkčných symbolov árnosti i a n_p^i predikátových symbolov árnosti i, a doména obsahuje zodpovedajúce počty m_o objektov, m_f^i funkcií árnosti i a m_p^i relácií i-tej árnosti medzi objektami, potom počet všetkých možných interpretácií je daný ako

$$(m_o)^{n_k} \left(\prod_{i=1,\dots} (m_f^i)^{n_f^i} \right) \left(\prod_{i=1,\dots} (m_p^i)^{n_p^i} \right)$$

kde $\prod_{i=1,\dots}$ označuje súčin.

Možné svety a modely

Možné svety na rozdiel od výrokovej logiky sú definované ako kombinácia interpretácie a domény. Doména nejakého sveta je neprázdna množina objektov (pri prázdnej množine by nebola možná žiadna interpretácia). Nad týmito objektami môžu byť definované doménové funkcie a relácie. Interpretácia špecifikuje, na ktoré prvky domény sveta odkazujú konštanty, predikátové a funkčné symboly, ktoré sú použité v logickej vete. Ten možný svet, v ktorom je nejaká logická veta pravdivá, sa označuje ako *model* danej vety.

Aj pri použití rovnakej interpretácie sa dva modely môžu líšiť vďaka svojim doménam – dve domény sa môžu napríklad líšiť v tých svojich prvkoch, ktoré nie sú referované použitou interpretáciou. Počet možných svetov ale aj modelov je takto neohraničený a preto pri predikátovej logike nie je možné (v analógii k metóde pravdivostných tabuliek pri výrokovej logike) použiť ich vymenovanie.

Predikátový počet 1. rádu

Nevýhodou syntaxe predikátového počtu nultého rádu je to, že umožňuje reprezentovať iba jednotlivé špecifické objekty, práca so skupinami objektov ani s anonymnými objektami nie je možná. Pre tento účel *predikátový počet prvého rádu* dopĺňa do svojej syntaxe premenné a kvantifikátory. Tieto doplnenia sú uvedené v Tab. 5.3.

	…	
< zložená veta >	::=	…
	|	< kvantifikátor > < premenná > < veta >
< term >	::=	… | < premenná >
	…	
< kvantifikátor >	::=	$\exists$ | $\forall$

Tab. 5.3: Syntax predikátového počtu prvého rádu (doplnenia voči nultému rádu)

Premenné

Premenné slúžia na reprezentáciu prvkov nejakej množiny objektov a syntakticky majú rovnakú úlohu ako konštanty a funkcie – môžu byť použité ako argumenty funkcií a predikátov. Tak napríklad v predikáte *dazd(X)* premenná X reprezentuje nejaké miesto s aktuálnym výskytom dažďa.

Univerzálny kvantifikátor

Na rozdiel od výrokovej logiky je možné vytvárať všeobecnejšie tvrdenia použitím predikátového počtu prvého rádu. Napríklad to, že každý človek má rád slnko, sa dá vyjadriť ako výraz

$$\forall X\ (cl(X) \to mr(X, slnko))$$

kde predikátový symbol *cl* vyjadruje vlastnosť "byť človekom" a predikát *mr* zase vzťah "mať rád". Podobne to, že každý človek má rád všetky zvieratá, by sa vyjadrilo pomocou výrazu

$$\forall X\ (cl(X) \to \forall Y\ (zv(Y) \to mr(X, Y)))$$

kde symbol *zv* vyjadruje vlastnosť "byť zvieraťom". Znak $\forall$ označuje *univerzálny (zovšeobecňovací) kvantifikátor* reprezentujúci "pre všetky hodnoty premennej". Veta tvaru "$\forall X < veta >$" je univerzálne kvantifikovaná. V prípade, že by výraz začínal viacerými kvantifikátormi, potom ich možno skrátene vyjadriť iba jedným, teda namiesto $\forall X\ \forall Y$ je možné použiť $\forall X, Y$ s tým istým významom.

Existenčný kvantifikátor

Nie vždy však platí nejaká vlastnosť univerzálne pre všetky objekty z nejakej množiny – napríklad nie každý človek má rád všetky zvieratá alebo každý človek má rád nejaké zviera avšak nie všetky. Toto by sa vyjadrilo

$$\exists X\ (cl(X) \wedge \exists Y\ (zv(Y) \wedge \neg mr(X,Y)))$$

s prvým významom alebo pomocou

$$\forall X\ (cl(X) \rightarrow \exists Y, Z\ (zv(Y) \wedge zv(Z) \wedge mr(X,Y) \wedge \neg mr(X,Z)))$$

s druhým významom. Znak $\exists$ označuje *existenčný kvantifikátor* reprezentujúci "existuje taká hodnota premennej, pre ktorú platí". Znamená to existenciu aspoň jednej vhodnej hodnoty, čo však nevylučuje, že tých hodnôt môže byť viac vrátane všetkých hodnôt. Veta tvaru "$\exists X < veta >$" je existenčne kvantifikovaná. V prípade, že by výraz začínal viacerými kvantifikátormi, potom ich možno skrátene vyjadriť iba jedným, teda namiesto $\exists X\ \exists Y$ je možné použiť $\exists X, Y$ s tým istým významom.

Dualita kvantifikátorov

Medzi obomi kvantifikátormi existuje vzťah na základe ekvivalentnosti, ktorý umožňuje zámenu jedného kvantifikátora za druhý (*dualita* kvantifikátorov). Je to

$$\exists X\ < veta > \equiv\ \neg\ (\forall X\ \neg < veta >)$$

prípadne v inom vyjadrení

$$\models \exists X\ < veta > \leftrightarrow\ \neg\ (\forall X\ \neg < veta >)$$

hovoriaci, že existencia výskytu takej hodnoty, pre ktorú veta platí, je ekvivalentná tomu, že nie je pravda, že veta neplatí pre žiadnu hodnotu danej premennej. Kvantifikátory sú navzájom zameniteľné, keď jeden z nich sa dá vyjadriť pomocou druhého podľa

$$\forall X\ p(X)\ \equiv\ \neg(\exists X\ \neg p(X)) \qquad \exists X\ p(X)\ \equiv\ \neg(\forall X\ \neg p(X))$$

Aj keď teda nie je nutné používať oba kvantifikátory a stačí použitie iba jedného z nich, kvôli čitateľnosti sa používajú oba súčasne.

Negácia kvantifikátorov

Z toho je zrejmé, že aplikovanie operátora negácie na vetu s kvantifikátorom dáva

$$\begin{aligned} \neg(\forall X\ p(X)) &\equiv \exists X\ \neg p(X) \\ \neg(\exists X\ p(X)) &\equiv \forall X\ \neg p(X) \end{aligned}$$

kde negovanie jedného typu kvantifikátora poskytuje kvantifikátor druhého typu s negáciou kvantifikovanej vety.

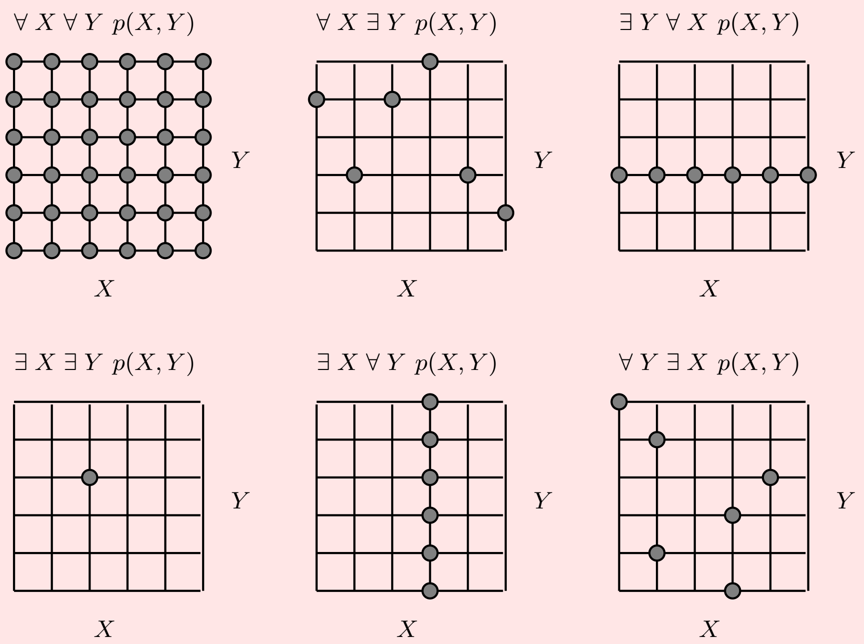
Dva kvantifikátory toho istého typu sú komutatívne (je možné zameniť ich poradie), platí teda

KOMUTATÍVNOSŤ KVANTIFIKÁTOROV

$$\begin{aligned} \forall X \, \forall Y \; p(X,Y) &\equiv \forall Y \, \forall X \; p(X,Y) \\ \exists X \, \exists Y \; p(X,Y) &\equiv \exists Y \, \exists X \; p(X,Y) \end{aligned}$$

avšak výskyt dvoch kvantifikátorov rôzneho typu nie je komutatívny, pretože vzťah platí iba jedným smerom

$$\exists X \, \forall Y \; p(X,Y) \models \forall Y \, \exists X \; p(X,Y)$$



Obr. 5.1: Kombinovanie kvantifikátorov

Vzťahy medzi kvantifikátormi je možné vizualizovať spôsobom ako na obr. 5.1. Z obrázku je očividné, akú úlohu hrá poradie jednotlivých kvantifikátorov a že toto poradie v prípade kombinovania rôznych typov kvantifikátorov nie je komutatívne.

Univerzálny kvantifikátor je distributívny nad konjunciou a existenčný kvantifikátor zase nad disjunkciou

DISTIBUTÍVNOSŤ KVANTIFIKÁTOROV

$$\begin{aligned} \forall X \; (p(X) \wedge q(X)) &\equiv \forall X \; p(X) \wedge \forall X \; q(X) \\ \exists X \; (p(X) \vee q(X)) &\equiv \exists X \; p(X) \vee \exists X \; q(X) \end{aligned}$$

avšak univerzálny kvantifikátor nie je distributívny nad disjunkciou a existenčný kvantifikátor nie je distributívny nad konjunkciou, pretože vzťahy platia iba jedným smerom

$$\begin{aligned} \forall\, X\ p(X) \lor \forall\, X\ q(X) &\models \forall\, X\ (p(X) \lor q(X)) \\ \exists\, X\ (p(X) \land q(X)) &\models \exists\, X\ p(X) \land \exists\, X\ q(X) \end{aligned}$$

Viazané a voľné premenné

Ak sa v logickej vete vyskytuje nejaká premenná a táto premenná je kvantifikovaná pomocou jedného z kvantifikátorov, tak sa označuje ako *viazaná* premenná. V opačnom prípade sa jedná o *voľnú* premennú. Pritom je možné, aby premenná hrala naraz obe úlohy. V príklade

$$\forall X\ (p(X, Y, Z) \to \exists Y\ q(X, Y, Z))$$

premenná X je viazaná (univerzálne) v celej vete, premenná Z je zase v celej vete voľná a premenná Y je voľná v ľavej časti vety zatiaľ čo v pravej časti je viazaná (existenčne). Voľná premenná je taká, ktorá môže byť nahradená ľubovoľnou inou voľnou premennou s tým, že oba tvary budú ekvivalentné. V trochu pozmenenom príklade

$$\forall X\ p(X, Y, Z) \to \exists Y\ q(X, Y, Z)$$

je premenná X viazaná iba v ľavej časti, v pravej časti nie je viazaná – je voľnou premennou. Znamená to, že v pravej časti ju možno nahradiť za novú premennú A s ekvivalentnou vetou

$$\forall X\ p(X, B, C) \to \exists Y\ q(A, Y, C)$$

Ak sa v logickej vete vyskytujú iba viazané premenné, potom takáto veta sa označuje za *uzavretú*. O splniteľnosti a validnosti hovoríme iba pri uzavretých vetách, pretože vďaka kvantifikácii premenných vieme tieto premenné interpretovať.

Interpretácia premennej

V prípade výskytu premenných sa situácia s interpretáciou viet trochu komplikuje. Za predpokladu, že vo vete vystupujú iba viazané premenné, tak sa vytvárajú *rozšírené interpretácie*, kde v každej je premenná interpretovaná ako jeden z objektov, na ktoré premenná odkazuje. Potom

- všeobecne kvantifikovaná veta je pravdivá v interpretácii I, ak je pravdivá pre všetky rozšírené interpretácie odvodené z I,
- existenčne kvantifikovaná veta je pravdivá v interpretácii I, ak je pravdivá pre aspoň jednu rozšírenú interpretáciu odvodenú z I.

Univerzálny kvantifikátor teda umožňuje formulovať tvrdenia o každom objekte, zatiaľ čo existenčný kvantifikátor umožňuje formulovať tvrdenia o nejakom objekte bez toho, aby tento objekt bol pomenovaný.

Príklad 5.3 Pre ilustráciu predpokladajme existenciu domény $\mathcal{D}$, ktorá obsahuje štyri objekty: *fero*, *jano*, *ondro* a *gusto*. Navyše obsahuje ešte jednu binárnu a jednu unárnu reláciu. Binárna relácia (označovaná *nema_rad* s významom, že objekt, ktorý je prvým argumentom, neobľubuje objekt, ktorý je druhým argumentom) je pravdivá pre dvojice objektov *jano-ondro*, *ondro-jano*, *fero-ondro*, *ondro-fero*, *jano-gusto* a *gusto-jano*. Pre ostatné možné dvojice je relácia nepravdivá. Unárna relácia (označovaná *kamarat*) je pravdivá pre objekty *gusto* a *ondro*, pre ostatné dva objekty je nepravdivá.

Skúsme vyjadriť tvrdenie "všetci nepriatelia môjho nepriateľa sú moji priatelia". Nech je to reprezentované vetou

$$\forall X \ (nepriatel(moj\_nepriatel, X) \rightarrow priatel(X))$$

s jednou konštantou a dvomi rôznymi predikátmi. Pre interpretáciu konštanty sú v danej doméne štyri rôzne možnosti, predikátový symbol *nepriatel* má iba jednu možnú interpretáciu a predikátový symbol *priatel* tiež iba jednu (pretože v doméne je iba jedna binárna a jedna unárna relácia). Jedna možná interpretácia (párujúca elementy danej vety s elementmi použitej domény) je

$$\{moj\_nepriatel^I = fero, nepriatel^I = nema\_rad, priatel^I = kamarat\}$$

ktorej prislúchajú štyri rozšírené interpretácie podľa Tab. 5.4. Z tabuľky

X	nepriatel(fero,X)	priatel(X)	nepriatel(fero,X) $\rightarrow$ priatel(X)
fero	FALSE	FALSE	TRUE
jano	FALSE	FALSE	TRUE
ondro	TRUE	TRUE	TRUE
gusto	FALSE	TRUE	TRUE

Tab. 5.4: Pravdivostná tabuľka pre rozšírené interpretácie

je zrejmé, že pre danú interpretáciu je celá veta interpretovaná ako pravdivá, lebo je pravdivá pre každú rozšírenú interpretáciu. Podobne to je aj pre prípad, že konštanta *moj_nepriatel* je interpretovaná ako objekt *jano*.

Na druhej strane, ak by táto konštanta bola interpretovaná či už ako *ondro* alebo *gusto*, tak celková veta by bola interpretovaná ako nepravdivá, pretože v týchto prípadoch by bola nepravdivá aspoň v jednej rozšírenej interpretácii.

Ak by doména obsahovala napríklad ešte jednu unárnu reláciu (označovanú *sportovec*), ktorá by bola pravdivá iba pre objekt *fero*, tak pre interpretáciu

$$\{moj\_nepriatel^I = fero, nepriatel^I = nema\_rad, priatel^I = sportovec\}$$

by boli možné ďalšie štyri rozšírené interpretácie, pri ktorých by daná veta bola interpretovaná ako nepravdivá – a to by sa nezmenilo ani ak by sa zmenila interpretácia konštanty. •

INTERPRETÁCIA PODĽA SYNTAKTICKEJ ZHODY

Je zrejmé, že interpretácia $priatel^I = kamarat$ je lepšia než $priatel^I = sportovec$ – avšak logika je formálny systém, ktorý nepracuje so sémantikou označení relácií a funkcií. Z toho dôvodu je vhodné (ak to je možné) v reprezentovaných vetách používať symboly rovnakého tvaru ako sú názvy zodpovedajúcich relácií a funkcií v uvažovanej doméne. Potom je možné prednostne uvažovať tie interpretácie, kde funkčné symboly a predikátové symboly sú interpretované na doménové funkcie a relácie s rovnakými názvami (interpretácia založená na *syntaktickej zhode mien*) – takáto interpretácia môže byť považovaná za tzv. *zamýšľanú* interpretáciu.

Bolo by to vhodné aj pri konštantách (ak by sme napríklad vedeli, ktorý konkrétny objekt spĺňa predstavu "môjho nepriateľa"). Nie vždy to je možné – niekedy proste vopred nie je známa vhodná interpretácia a nezostáva nič iné ako použiť konštantu, ktorej meno nezodpovedá priamo žiadnemu objektu z domény (príkladom takéhoto prístupu je konštanta *moj_nepriatel*) alebo použiť premennú, ak je možnosť jej vhodnej kvantifikácie. Na základe tohto by sa dala veta "všetci nepriatelia môjho nepriateľa sú moji priatelia" pre danú doménu (s ohľadom na mená v nej použité) pretransformovať do tvaru

$$\exists Y\ (\neg kamarat(Y) \wedge \forall X\ (nema\_rad(Y, X) \rightarrow kamarat(X)))$$

kde namiesto konštanty sa použila forma premennej s vlastnosťou, ktorá ju definuje.

Príklad 5.4 Podobne aj v ilustračnom príklade na strane 130 v reprezentácii problému sa používajú rovnaké názvy ako v slovnom popise toho problému (*strasidlo*, *priepast*, *zapach*, *prievan*, atď.).

Medzi pozíciou, ktorá je reprezentovaná jedným poľom (miestnosťou) v danom svete, a umiestnením tejto pozície podľa dvoch súradníc existuje závislosť, ktorú je možné modelovať ako štrukturálny vzťah medzi príslušnými objektmi. Vďaka tomuto nie je potrebné uvažovať jednotlivé polia a aj hodnoty súradníc ako samostatné objekty, ale je možné jeden typ objektu odvodiť z iného typu objektov. Preto polia nepoužívajú explicitné pomenovania ale sú vyjadrené binárnou funkciou s príslušným funkčným symbolom *pole*. Pomenovanými objektami budú indexy miestností 1, 2, 3 a 4.

Takto implicitne vyjadrené polia sú nositeľmi vlastností: *priepast* (na danom poli sa nachádza priepasť), *strasidlo* (na danom poli je umiestnené strašidlo), *prievan* (na danom poli cítiť prievan) a *zapach* (na danom poli cítiť zápach). To, či nejaké pole má tú ktorú vlastnosť, je reprezentované príslušným unárnym predikátom.

V popise zákonitostí ilustračného sveta sa vyskytuje fráza "nejaké pole". Keďže nie je jasné, o ktoré konkrétne pole sa jedná, bola zvolená reprezentácia pomocou premennej. Keďže je domnienka, že vlastnosti č. 2 (ak je na nejakom poli strašidlo, tak na susedných poliach je cítiť zápach) a č. 3 (ak je na nejakom poli priepasť, tak na susedných poliach je cítiť prievan) sú všeobecne platné pre každé pole, tak príslušná premenná je univerzálne kvantifikovaná.

Medzi poliami existujú závislosti, keď vlastnosť jedného poľa súvisí s vlastnosťou susedného poľa (napr. *priepast* a *prievan*). Ak by sa z výskytu priepasti na nejakom poli odvádzal prievan na susednom poli, tak potom susedné pole by bolo reprezentované premennou s univerzálnou kvantifikáciou (pretože daná vlastnosť platí pre každé zo susedných polí).

$$\forall\, S\ (priepast(S) \to \forall\, R\ (susedne(S,R) \to prievan(R)))$$

Vzťah je iba jednosmerný, z existencie prievanu na všetkých susedných poliach nevyplýva existencia priepasti na danom poli. Naopak, ak by sa z prievanu na nejakom poli odvádzal výskyt priepasti na susednom poli, tak potom by susedné pole bolo reprezentované premennou s existenčnou kvantifikáciou (vlastnosť nemusí platiť pre všetky susedné polia, avšak platí aspoň pre jedno z nich).

$$\forall\, S\ (prievan(S) \leftrightarrow \exists\, R\ (susedne(S,R) \land priepast(R)))$$

Na záver je ešte potrebné vyjadriť pojem susednosti polí, ktorý je založený na následnosti dvoch čísel. Keďže v predikátovej logike žiadna vlastnosť následnosti či usporiadania čísel nie je definovaná, je ju nutné explicitne reprezentovať pomocou predikátu, definujúceho úplné usporiadanie objektov, vyjadrujúcich hodnoty súradníc. Použitý predikát *po* vyjadruje, ktorý objekt nasleduje za ktorým objektom, teda napríklad *po(1,2)* reprezentuje fakt, že po čísle 1 nasleduje číslo 2. •

CNF PRE PREDIKÁTOVÚ LOGIKU

Aj v predikátovej logike sa používa CNF. Opäť je to konjunkcia klauzúl v tvare disjunkcií literálov. Avšak teraz literál namiesto symbolu obsahuje predikát buď v priamej alebo negovanej podobe. Definícia CNF pre oblasť predikátovej logiky je v Tab. 5.5.

$$\begin{array}{rcl}
<\text{veta}> & ::= & <\text{klauzula}> \mid <\text{klauzula}> \wedge <\text{veta}> \\
<\text{klauzula}> & ::= & <\text{literál}> \mid <\text{literál}> \wedge <\text{klauzula}> \\
<\text{literál}> & ::= & <\text{predikát}> \mid \neg <\text{predikát}> \\
<\text{predikát}> & ::= & <\text{predikátový symbol}>(<\text{term}>*) \\
<\text{term}> & ::= & <\text{konštanta}> \\
 & \mid & <\text{funkcia}> \\
 & \mid & <\text{premenná}> \\
<\text{funkcia}> & ::= & <\text{funkčný symbol}>(<\text{term}>+)
\end{array}$$

Tab. 5.5: CNF pre predikátovú logiku

Ako vidno, transformácia do takejto formy musí nielen nahradiť binárne operátory s výnimkou $\wedge$ a $\vee$ (a tie usporiadať tak, aby sa disjunkcia vyskytovala v rámci konjunkcie a nie naopak) a negáciu vnoriť na úroveň literálov, ale aj odstrániť oba typy kvantifikátorov. Ak vo výslednom tvare literály obsahujú premenné, potom sa jedná o také premenné, ktoré boli univerzálne kvantifikované pomocou zovšeobecňovacieho kvantifikátora.

PREVOD DO CNF

Ľubovoľná veta v predikátovom počte prvého rádu môže byť transformovaná na *inferenčne ekvivalentnú* vetu vyjadrenú v tvare CNF – CNF veta bude nesplniteľná práve vtedy, ak pôvodná veta je nesplniteľná. Algoritmus pre túto transformáciu je rozšírením algoritmu Alg. 1.1 – po rozšírení pozostáva zo siedmich krokov (uvedených v Alg. 5.1), pričom štyri z nich (kroky 1, 2, 3 a 7) majú rovnaký cieľ a sú rovnako realizovateľné ako pri výrokovom počte. Preto sa zameriame iba na tri doplnené kroky.

ŠTANDARDIZÁCIA PREMENNÝCH

Cieľom štandardizácie premenných je zabezpečiť, aby žiadne dve rôzne premenné neboli pomenované rovnako. Ilustrujme to na dvoch jednoduchých ukážkach

$$\forall X\ p(X) \vee \exists X\ q(X) \qquad\qquad \forall X\ (p(X) \rightarrow \exists X\ q(X))$$

kde v oboch prípadoch premenná X, hrajúca úlohu argumentu predikátu p, je rôzna od premennej X, ktorá vystupuje ako argument predikátu q. Keďže sa nepoužívajú voľné premenné, tak každá premenná je kvantifikovaná a teda by jej mal zodpovedať práve jeden kvantifikátor, ktorý ju

vstup: veta F v ľubovoľnom tvare
výstup: veta F v CNF tvare

1. $F :=$ *eliminácia_ekvivalencií(F)*
2. $F :=$ *eliminácia_implikácií(F)*
3. $F :=$ *vnorenie_negácií(F)*
4. $F :=$ *štandardizácia_premenných(F)*
5. $F :=$ *vylúčenie_existenčných_kvantifikátorov(F)*
6. $F :=$ *vylúčenie_zovšeobecňovacích_kvantifikátorov(F)*
7. $F :=$ *úprava_na_CNF(F)*

Alg. 5.1: Transformácia do CNF

uvádza. V prípade, že pre jedno meno premennej existuje viac kvantifikátorov, potom viac premenných používa rovnaké meno – toto meno v rámci pôsobnosti nejakého kvantifikátora označuje nejakú premennú, zatiaľ čo v rámci pôsobnosti iného kvantifikátora označuje inú premennú – prípad typu $\forall X\ p(X) \lor \exists X\ q(X)$. Ak sa premenná nachádza v rámci pôsobnosti viacerých kvantifikátorov (teda kvantifikátory sú vnorené), tak premenná prináleží k tomu kvantifikátoru, ktorý je k nej zľava bližšie – prípad typu $\forall X\ (p(X) \to \exists X\ q(X))$.

Keďže na mene premennej nezáleží a platí, že dve vety, ktoré sa líšia iba názvami premenných, sú ekvivalentné, tak pre účely rozoznávania premenných je potrebné ich premenovať tak, aby žiadne dve premenné nepoužívali rovnaké meno. Naše ukážky sa tak môžu transformovať napríklad na tvar

$$\forall X\ p(X) \lor \exists Y\ q(Y) \qquad\qquad \forall Z\ (p(Z) \to \exists X\ q(X))$$

Vylúčenie $\exists$ kvantifikátora

Výskyt existenčného kvantifikátora vo výraze $\exists X\ kamarat(X)$ hovorí, že existuje aspoň jeden taký objekt v doméne $\mathcal{D}$, pre ktorý je unárny predikát *kamarat* pravdivý (z už uvedeného príkladu tejto domény na strane 139 vieme, že to sú objekty *ondro* a *gusto*). To znamená, že pravdivosť výrazu sa zachová, ak by sme premennú X nahradili konštantou, ktorá je interpretovaná na jeden z týchto objektov – napríklad by sa použil tvar $kamarat(ondro)$

Iná situácia by bola, ak by sme nevedeli, pre ktorý z objektov je daný predikát pravdivý – jediné čo vieme je to, že taký objekt existuje. Je možné tento neznámy objekt pomenovať pomocou novej konštanty, ktorá sa doposiaľ nevyskytla (aby nedošlo ku konfliktu s nejakou už existujúcou konštantou), a všetky výskyty existenčne kvantifikovanej premennej nahradiť touto konštantou. Takto je možné výraz $\exists X\ kamarat(X)$ nahradiť pomocou $kamarat(niekto)$, kde $niekto$ je nová konštanta. Takáto konštanta sa označuje ako *skolemova konštanta* a celý proces náhrady sa nazýva *skolemizácia*.

SKOLEMOVA KONŠTANTA

Zložitejšia situácia nastáva, ak existenčný kvantifikátor sa nachádza v oblasti pôsobenia zovšeobecňovacieho kvantifikátora ako pri

$$\forall Y\ \exists X\ (kamarat(X) \wedge nema\_rad(Y, X))$$

pretože teraz sa náhrada premennej X deje v kontexte, ktorý je daný univerzálne kvantifikovanou premennou Y. Pôvodný význam vety je, že pre každý objekt existuje nejaký kamarát – pritom pre dva rôzne objekty sa môže jednať vždy o iného kamaráta. Toto by sa stratilo, pretože po náhrade premennej X za konštantu $niekto$ by sa význam zmenil tak, že pre dva rôzne objekty sa bude jednať o rovnakého kamaráta.

SKOLEMOVA FUNKCIA

Preto sa v tomto prípade namiesto skolemovej konštanty použije *skolemova funkcia*, ktorá je funkciou tej premennej, ktorá vytvára kontext[1]. Podobne ako pri konštante, ani teraz funkčný symbol nesmie vytvárať kolízie. Pri takejto náhrade by bol výsledok

$$\forall Y\ (kamarat(niekto(Y)) \wedge nema\_rad(Y, niekto(Y)))$$

Môže nastať prípad, keď existenčný kvantifikátor je v oblasti pôsobenia viacerých zovšeobecňovacích kvantifikátorov, ako napríklad

$$\forall Z\ \forall Y\ \exists X\ (\neg nema\_rad(Z, X) \vee \neg nema\_rad(X, Y))$$

V takomto prípade skolemova funkcia bude mať viac argumentov – pre daný prípad premenná X bude nahradená binárnou funkciou $niekto(Z, Y)$, ktorá zohľadní úplný kontext daný dvomi kvantifikátormi.

VYLÚČENIE $\forall$ KVANTIFIKÁTORA

Zovšeobecňovacie kvantifikátory je možné posunúť úplne na začiatok vety. Umožňujú to tieto pravidlá (predpokladá sa, že predikát q neobsahuje premennú X v rámci svojich argumentov)

$$\begin{array}{lcl}
\forall X\ p(X, ...) \vee q(...) & \rightarrow & \forall X\ (p(X, ...) \vee q(...)) \\
q(...) \vee \forall X\ p(X, ...) & \rightarrow & \forall X\ (q(...) \vee p(X, ...)) \\
\forall X\ p(X, ...) \wedge q(...) & \rightarrow & \forall X\ (p(X, ...) \wedge q(...)) \\
q(...) \wedge \forall X\ p(X, ...) & \rightarrow & \forall X\ (q(...) \wedge p(X, ...))
\end{array}$$

[1] Skolemovu konštantu možno chápať ako skolemovu funkciu s nulovou árnosťou.

Po vykonaní posunu je výsledkom tvar, kde všetky kvantifikátory sú nasledované konjunkciou klauzúl, ktorá kvantifikátory neobsahuje. A keďže všetky premenné, ktoré sa vo vete vyskytujú, sú všeobecne kvantifikované, je možné kvantifikátory jednoducho odstrániť a ponechať iba zvyšnú časť vety.

Príklad 5.5 Ako príklad pre transformáciu do CNF uvažujme nasledujúcu časť reprezentácie z ilustračného príkladu na strane 130

$$\forall\, S\ (prievan(S) \leftrightarrow \exists\, R\ (susedne(S,R) \wedge priepast(R)))$$

Z tejto vety po odstránení operátorov ekvivalencie a implikácie získame

$$\begin{aligned}\forall\, S\ (\quad & (\neg prievan(S) \vee \exists\, R\ (susedne(S,R) \wedge priepast(R)))\ \wedge \\ & (\neg\exists\, R\ (susedne(S,R) \wedge priepast(R)) \vee prievan(S))\ )\end{aligned}$$

a po vnorení negácie bude mať veta nasledovný tvar

$$\begin{aligned}\forall\, S\ (\quad & (\neg prievan(S) \vee \exists\, R\ (susedne(S,R) \wedge priepast(R)))\ \wedge \\ & (\forall\, R\ (\neg susedne(S,R) \vee \neg priepast(R)) \vee prievan(S))\ )\end{aligned}$$

Pretože dve rôzne premenné kvantifikované dvomi kvantifikátormi používajú rovnaké meno, je potrebné jednu z nich premenovať

$$\begin{aligned}\forall\, S\ (\quad & (\neg prievan(S) \vee \exists\, R\ (susedne(S,R) \wedge priepast(R)))\ \wedge \\ & (\forall\, Q\ (\neg susedne(S,Q) \vee \neg priepast(Q)) \vee prievan(S))\ )\end{aligned}$$

Vo vete sa vyskytuje jedna existenčne kvantifikovaná premenná. Túto premennú je potrebné nahradiť pomocou skolemovej funkcie, čo umožní odstránenie existenčného kvantifikátora

$$\begin{aligned}\forall\, S\ (\quad & (\neg prievan(S) \vee (susedne(S,f(S)) \wedge priepast(f(S))))\ \wedge \\ & (\forall\, Q\ (\neg susedne(S,Q) \vee \neg priepast(Q)) \vee prievan(S))\ )\end{aligned}$$

Po presunutí zovšeobecňovacích kvantifikátorov je možné ich odstrániť a záverečne ešte vetu upraviť na tvar CNF

$$\begin{aligned} & (\neg prievan(S) \vee susedne(S,f(S))) \\ \wedge\quad & (\neg prievan(S) \vee priepast(f(S))) \\ \wedge\quad & (\neg susedne(S,Q) \vee \neg priepast(Q) \vee prievan(S))\end{aligned}$$

•

Rozširovanie bázy znalostí

Všetky klauzuly, ktoré boli odvodené pre reprezentáciu nejakej úlohy, vytvárajú spolu znalostnú bázu KB danej úlohy (teda znalostná báza má tvar konjunkcie klauzúl). Na základe tejto znalostnej bázy je možné *dedukovať* ďalšie znalosti[2], opäť vo forme klauzúl, ktoré je následne možné do tejto bázy pridať. Celý proces sa tak môže opakovať až dovtedy, keď už deduktívne odvodzovanie nie je schopné vyprodukovať znalosť, ktorá by ešte nebola známa.

Príklad 5.6 Tak napríklad v ilustračnom príklade na strane 130 je možné na základe vnemov získaných na bezpečnej pozícii $pole(1, 1)$ (absencia prievanu aj zápachu) do znalostnej bázy pridať znalosti $\neg prievan(pole(1, 1))$ a $\neg zapach(pole(1, 1))$ a na základe toho odvodiť nové znalosti o priepastiach $\neg priepast(pole(1, 2))$ a $\neg priepast(pole(2, 1))$, ako aj nové znalosti o strašidlách $\neg strasidlo(pole(1, 2))$ a $\neg strasidlo(pole(2, 1))$, čo znamená bezpečnosť polí $pole(1, 2)$ a $pole(2, 1)$.

Po presune doprava na bezpečnú pozíciu $pole(2, 1)$ sa zaznamená zmyslový vnem, získaný na tomto poli, ako nové znalosti $prievan(pole(2, 1))$ a $\neg zapach(pole(2, 1))$ a následne sa odvodí neprítomnosť strašidla v okolí na pozíciách $pole(2, 2)$ a $pole(3, 1)$.

Po presune na ďalšiu známu bezpečnú pozíciu $pole(1, 2)$ a zaznamenaní vnemov $\neg prievan(pole(2, 1))$ a $zapach(pole(2, 1))$ je už možné odvodiť prítomnosť strašidla na pozícii $pole(1, 3)$ a súčasne neprítomnosť priepasti na tomto poli a taktiež na pozícii $pole(2, 2)$ – a teda prítomnosť priepasti na pozícii $pole(3, 1)$ a súčasne bezpečnosť pozície $pole(2, 2)$.

Takže je možné sa presunúť na ďalšie pole, o ktorom sa vie že je bezpečné, a skúmanie sveta môže pokračovať. •

Logické vyplývanie

Odvodzovanie je založené na pojme logického vyplývania, kde nejaká veta P zahŕňa nejakú inú vetu Q (značené ako $P \models Q$). Je zrejmé, že úlohu P hrá znalostná báza KB obsahujúca známe informácie a úlohu Q zase nová znalosť, ktorá môže byť odvodená z KB. Dôkaz zahrnutia jednej vety druhou je možné v zásade realizovať tromi rôznymi spôsobmi:

- kontrolou interpretácií,
- priamym dôkazom,
- dôkazom sporom.

[2] Prostredníctvom riešenia dedukčnej úlohy ako už bolo spomenuté v kapitole venovanej inferencii vo výrokovej logike.

KONTROLA INTERPRETÁCIÍ

Pri prvom spôsobe je potrebné skontrolovať všetky možné interpretácie, vybrať z nich tie, pri ktorých je P interpretovaná ako pravdivá a skontrolovať, či v každej takejto interpretácii je aj Q interpretovaná ako pravdivá. Jedná sa o ideovo jednoduchý spôsob, avšak z pohľadu použitia je vďaka veľkému množstvu možných interpretácií táto možnosť viac teoretická než praktická.

PRIAMY DÔKAZ

Pri *priamom dôkaze* sa používajú rozličné odvodzovacie pravidlá v tvare ekvivalentných transformácií (napr. modus ponens), umožňujúce transformovať logické vety na ekvivalentné tvary. Dôkaz má potom podobu výberu a uplatňovania takýchto pravidiel na vetu P dovtedy, až sa táto zo svojej počiatočnej podoby pretransformuje na tvar, ktorý bude obsahovať Q. Pri tomto spôsobe je problematickou realizácia voľby, ktoré pravidlo použiť na ktorú časť vety v ktorom transformačnom kroku. Keďže táto voľba sa obtiažne realizuje "cieľovo orientovaným" spôsobom, počet možností nefavorizuje ani tento spôsob.

DÔKAZ SPOROM

Dôkaz sporom je založený na dvoch pojmoch: validnosti a splniteľnosti. Ich význam je pre predikátovú logiku rovnaký ako pre logiku výrokovú. Na základe využitia týchto pojmov je možné pre dôkaz platnosti nejakej vety Q (označovanej aj ako cieľová znalosť) odvodiť

$$
\begin{aligned}
P \models Q &\leftrightarrow valid(P \rightarrow Q) \\
valid(P \rightarrow Q) &\leftrightarrow valid(\neg P \vee Q) \\
valid(\neg P \vee Q) &\leftrightarrow nespl(\neg(\neg P \vee Q)) \\
nespl(\neg(\neg P \vee Q)) &\leftrightarrow nespl(P \wedge \neg Q)
\end{aligned}
$$

kde *valid(F)* znamená, že veta F je validná, a *nespl(F)* zase že veta F je nesplniteľná. Z uvedeného vyplýva, že pre dôkaz platnosti nejakej novej cieľovej znalosti na základe danej znalostnej bázy (teda že znalostná báza KB zahŕňa cieľovú znalosť resp. že cieľová znalosť logicky vyplýva zo znalostnej bázy) stačí dokázať, že súčasne nemôže platiť znalostná báza a aj negácia cieľovej znalosti. Znalostná báza (reprezentovaná ako konjunkcia klauzúl) sa jednoducho rozšíri pridaním jednej alebo viacerých klauzúl, reprezentujúcich negáciu dokazovanej cieľovej znalosti. Ak sa dokáže neplatnosť takto rozšírenej znalostnej bázy (teda pridaná negácia cieľovej znalosti je v rozpore so známymi axiómami reprezentovanými pôvodnou znalostnou bázou), tak to znamená platnosť cieľovej znalosti.

REZOLVENCIA

Pri tomto dôkaze sa používa *rezolvenčná metóda*, ktorá je populárnou metódou v oblasti automatického dokazovania. Jej popularita je založená na tom, že môže byť ľahko automatizovaná, pretože je založená na používaní iba jedného odvodzovacieho pravidla – *rezolvenčného odvodzovacieho pravidla*. Toto pravidlo produkuje rezolventy na základe dvoch vstupných klauzúl – tieto vstupné klauzuly musia obsahovať dvojicu *komplementárnych literálov* (každý z nich pochádza z inej vstupnej klauzuly). Na rozdiel od výrokovej logiky, kde úlohu komplementárnych literálov hrali symbol a jeho negácia, pri predikátovej logike sa jedná o literály založené na predikátoch – jeden z nich je vždy priamo vyjadreným literálom a druhý sa zhoduje s negáciou prvého literálu.

ZHODA LITERÁLOV

Otázkou je, čo pojem "zhodovania sa" pri definícii komplementárnych literálov vlastne reprezentuje. Odpoveď závisí od toho, aká logika bola použitá pre reprezentáciu znalostí. Ak sú použité iba predikátové symboly (teda árnosť všetkých predikátov je nulová – takáto logika je ekvivalentná výrokovej logike), potom pod zhodnosťou sa myslí rovnaké pomenovanie symbolu. V tomto prípade pod komplementárnymi literálmi sa rozumie dvojica predikát (plne určený svojím predikátovým symbolom) a jeho negácia.

Zložitejšia situácia nastáva, keď v predikátoch sú použité aj termy vrátane premenných. V takomto prípade intuitívne je možné za komplementárne literály považovať trebárs dvojicu $p(q, f(r))$ a $\neg p(q, f(r))$, zatiaľ čo na druhej strane zase dvojica $p(s, g(r))$ a $\neg p(g(s), r)$ nebude považovaná za komplementárne literály. Avšak čo napríklad pri dvojici $p(X, f(X))$ a $\neg p(g(Z), Y)$, kde X, Y a Z reprezentujú premenné?

SUBSTITÚCIA TERMOV

Toto rieši *substitúcia termov* za premenné v literáloch. Premenné, ktoré sa vyskytujú vo vete, ktorá je v CNF tvare, sú všeobecne kvantifikované. Náhradou takejto premennej napríklad za konštantu dochádza k znižovaniu počtu uvažovaných interpretácií (namiesto mnohých rozšírených interpretácií bola zvolená iba jedna). To však nevadí, pretože cieľom je realizovať dôkaz sporom – a nepravdivosť v jednej rozšírenej interpretácii znamená celkovú nepravdivosť vďaka univerzálnej kvantifikácii premennej.

UNIFIKÁCIA

Cieľom *unifikácie* je nájsť takú substitúciu, aby dve vety boli ekvivalentné. Potom pod zhodnosťou dvoch predikátov sa rozumie situácia, keď tieto dva predikáty sú navzájom unifikovateľné. Formálne pod unifikáciou dvoch viet P a Q sa rozumie taká operácia

$$unify(P, Q) = \theta$$

ktorej výsledkom je *unifikátor* θ, umožňujúci dosiahnuť rovnakú podobu oboch viet v prípade, že na obe bude aplikovaná substitúcia daná týmto unifikátorom, a teda bude platiť

$$subst(\theta, P) = subst(\theta, Q)$$

UNIFIKÁTOR

Unifikátor je vlastne množina substitúcií premenných, pričom pre každú premennú môže obsahovať najviac jednu náhradu. V prípade ilustračných viet $p(X, f(X))$ a $p(g(Z), Y)$ by unifikátor mohol mať trebárs jeden z tvarov

$$\theta = \{X/g(Z), Y/f(g(Z))\} \qquad \theta = \{X/g(Z), Y/f(g(Z)), Z/q\}$$

kde zápis a/b znamená, že pri substitúcii s použitím daného unifikátora je a nahrádzané pomocou b (teda napríklad v druhom prípade premenná Z bola nahradená konštantou q). Je zrejmé, že úloha nájdenia unifikátora nemusí poskytnúť jednoznačné riešenie – principiálne existuje viacero možných unifikátorov. Tieto unifikátory za navzájom líšia na základe svojej všeobecnosti – všeobecnejší je ten z nich, ktorý menej obmedzuje hodnoty premenných (teda v predchádzajúcej ukážke ľavý unifikátor je všeobecnejší, pretože na rozdiel od pravého neobmedzuje hodnotu premennej Z). Aby sme sa vyhli nejednoznačnosti, tak sa zvykne požadovať nájdenie *najvšeobecnejšieho* unifikátora.

Pre proces unifikácie v zásade platia dve pravidlá, ktoré určujú prípady, keď unifikácia nie je možná. Sú to

PROCES UNIFIKÁCIE

- konštanta ani funkcia nemôžu byť nahradené premennou,
- premenná nemôže byť nahradená termom, ktorý obsahuje danú premennú,

a teda unifikátor nemôže obsahovať substitúciu typu p/X resp. $f(p)/X$ ani $X/f(X)$, kde p reprezentuje konštantu, f funkčný symbol a X zase premennú.

Toto je obsiahnuté v unifikačnom algoritme Alg. 5.2. Algoritmus začína s prázdnym unifikátorom, pričom postupne unifikované výrazy rozkladá na jednoduchšie prvky a tie sa pokúša navzájom unifikovať. V prípade, že unifikácia na nejakom mieste skončí neúspechom (reprezentovaným pomocou symbolu $\otimes$), tak aj celkový pokus o unifikáciu je neúspešný a nemá zmysel si uchovať dovtedy vybudovaný unifikátor. Pokiaľ unifikované objekty sú zložené (teda sú predikátmi alebo funkciami), tak sa najprv unifikujú ich symboly a nato zoznamy ich argumentov. Ak unifikované objekty sú zoznamami (reprezentujúcimi zoznamy argumentov funkcií alebo predikátov), tak sa unifikácia vykonáva rekurzívne – najprv sú unifikované prvé prvky zoznamov a následne sa unifikujú zvyšky týchto zoznamov bez prvých prvkov. Ak niektorý unifikovaný prvok je premennou (pre testovanie sa používa *var?*), tak algoritmus prejde na špeciálnu sekciu pre unifikáciu premennej. A ak ani jedna z uvedených možností nie je vhodná, znamená to že dané prvky sa nedajú unifikovať a signalizuje sa neúspech.

UNIFIKAČNÝ ALGORITMUS

V časti pre unifikáciu premennej sa pred vytvorením substitúcie vykonáva niekoľko kontrol. Najprv sa kontroluje, či sa daná premenná už skôr

UNIFIKÁCIA PREMENNEJ

vstup: predikáty x a y
výstup: unifikátor θ v prípade úspechu alebo $\oslash$ pri neúspechu

```
unify(x, y)
    θ := {}
    return unify_aux(x, y, θ)

unify_aux(x, y, θ)
    if θ = ⊘ then return ⊘
    else if x = y then return θ
    else if var?(x) then return unify_var(x, y, θ)
    else if var?(y) then return unify_var(y, x, θ)
    else if x = a(p, ..., r) and y = b(s, ..., u) then return
            unify_aux([p, ..., r], [s, ..., u], unify_aux(a, b, θ))
    else if x = [p, q, ..., r] and y = [s, t, ..., u] then return
            unify_aux([q, ..., r], [t, ..., u], unify_aux(p, s, θ))
    else return ⊘

unify_var(var, x, θ)
    if var/y ∈ θ then return unify_aux(y, x, θ)
    else if x/y ∈ θ then return unify_aux(var, y, θ)
    else if var ∈ x then return ⊘
    else return θ = θ ∪ {var/x}
```

Alg. 5.2: Unifikačný algoritmus

nevyskytla v nejakej substitúcii – ak áno, tak sa tá substitúcia použije (pretože nie je možné pre premennú vytvoriť viac substitúcií). Keďže druhým argumentom pre *unify_var* môže byť tiež premenná, tak sa toto skúmanie robí aj pre tento argument. A záverečne sa kontroluje, či sa premenná (prvý argument) nevyskytuje ako súčasť druhého argumentu (to by došlo k porušeniu jedného z uvedených pravidiel pre unifikáciu). Ak všetky tieto kontroly sú v poriadku, tak sa vytvorí nová substitúcia a zaradí sa do unifikátora.

V prípade, že pri rezolvencii vystupujú klauzuly obsahujúce aj premenné, je nutné pre dosiahnutie komplementarity dvoch literálov nájsť vhodný unifikátor, ktorý vhodným spôsobom obmedzí premenné v týchto dvoch literáloch. Toto obmedzenie sa musí preniesť aj na výslednú rezolventu (pretože je rezolventou iba za platnosti obmedzenia daného nájdeným unifikátorom). Rezolvenčné odvodzovacie pravidlo bude mať potom o niečo všeobecnejší tvar než pri výrokovej logike

Zovšeobecnená rezolvencia

$$\frac{P \vee Q \qquad \neg R \vee S}{subst(\theta, P \vee S\ )}$$

kde unifikátor θ je taký, že $subst(\theta, Q) = subst(\theta, R)$.

Výsledná rezolventa nesmie obsahovať viacnásobný výskyt literálu. Pri výrokovej logike to bolo jednoducho identifikovateľné – kontrolovalo sa, či nejaký literál sa v rezolvente nevyskytuje viackrát. V prípade zovšeobecnenej rezolvencie pri predikátovej logike môže nastať zložitejší prípad, keď rezolventa obsahuje dva výskyty predikátu, ktoré však nie sú identické ale líšia sa svojimi argumentami. V tomto prípade je potrebné oba výskyty unifikovať, nahradiť ich unifikovanou podobou a samozrejme nájdený unifikátor aplikovať na celú rezolventu.

Redundantný výskyt literálu

Príklad 5.7 Rezolvencia klauzuly $q(X) \vee p(X, a) \vee r(X)$ s inou klauzulou $\neg r(f(Y)) \vee p(f(b), Z)$ má tvar

$$\frac{q(X) \vee p(X, a) \vee r(X) \qquad \neg r(f(Y)) \vee p(f(b), Z)}{subst(\theta = \{X/f(Y)\}, q(X) \vee p(X, a) \vee p(f(b), Z))}$$

kde rezolventa po aplikácii nájdeného unifikátora $\theta = \{X/f(Y)\}$ nadobudne tvar $q(f(Y)) \vee p(f(Y), a) \vee p(f(b), Z)$. Keďže obe verzie predikátu p sú unifikovateľné, výsledná rezolventa bude mať tvar $q(f(b)) \vee p(f(b), a)$ vďaka unifikácii $\{Y/b, Z/a\}$.

•

Špeciálnym prípadom je situácia, keď v oboch vstupných klauzulách sa vyskytuje premenná rovnakého mena. Dôvodom je to, že platí ekvivalentnosť

Kolízia premenných

$$\forall X\ (p(X) \wedge g(X)) \ \equiv\ \forall X\ \forall Y\ (p(X) \wedge g(Y))$$

Keďže premenné, ktoré sa nachádzajú v klauzulách vety v tvare CNF sú implicitne univerzálne kvantifikované, tak potom dôsledkom je, že

- literály $p(X, f(r))$ a $\neg p(g(r), X)$ sú komplementárne,
- rezolventou klauzúl $p(X) \vee q(r)$ a $\neg q(X)$ nie je $p(r)$,

- rezolventou klauzúl $p(X) \vee q(r)$ a $\neg q(r) \vee s(X)$ nie je $p(X) \vee s(X)$,

pretože ak v druhej vstupnej klauzule sa zmení označenie premennej z X na Y, potom pre prvý prípad bude existovať unifikátor $\theta = \{X/g(r), Y/f(r)\}$, v druhom prípade rezolventou bude $p(X)$ a v treťom prípade bude rezolventou klauzula $p(X) \vee s(Y)$.

Preto je ešte pred samotnou rezolvenciou potrebné premenovať premenné tak, aby nenastal prípad výskytu rovnakého mena premennej v oboch vstupných klauzulách. Alternatívne to je možné zabezpečiť ako dodatočný krok č. 8 v algoritme Alg. 5.1 pre transformáciu viet do tvaru CNF – v klauzulách sa premenujú premenné tak, aby v žiadnej z dvojíc klauzúl sa nevyskytovalo rovnaké meno premennej.

ALGORITMUS PRE DÔKAZ SPOROM

Formálna podoba algoritmu pre dôkaz sporom pomocou použitia rezolvenčnej metódy je rovnaká ako pre výrokovú logiku (Alg. 2.1). Jediný rozdiel je ten, že vstupom je KB, ktorá už obsahuje aj negáciu dokazovanej hypotézy (teda $KB = KB \wedge \neg Q$).

Príklad 5.8 Ako ukážku rezolvenčného uvažovania opäť zoberme ilustračný príklad na strane 130 s tým, že chlapci sa nachádzajú na pozícii $pole(1, 1)$ a na tejto pozícii nevnímajú ani prievan ani zápach. Skúšajú zistiť, či pozícia $pole(1, 2)$ je bezpečná, teda či sa na danej pozícii dá vylúčiť prítomnosť strašidla aj priepasti. Pre vylúčenie priepasti si stanovia hypotézu $\neg priepast(pole(1, 2))$, ktorú sa pokúsia dokázať. Hypotézu teda negujú a snažia sa v dôkaze získať prázdnu klauzulu.

Derivačný graf je zobrazený na Obr. 5.2, pričom sú zobrazené iba tie rezolvencie, ktoré prispievajú k redukcii na prázdnu klauzulu (a teda vďaka tomu má graf stromovú štruktúru). Uzly grafu reprezentujú klauzuly dvojakého typu:

- vstupné klauzuly (klauzuly vybraté zo znalostnej bázy),
- rezolventy.

Prvý typ klauzúl je znázornený plnými uzlami, ktoré nemajú predchodcov, zatiaľ čo čiarkované uzly s predchodcami reprezentujú druhý typ klauzúl. Hrany reprezentujú uskutočnené rezolvencie, keď vždy dve hrany spájajú vstupné klauzuly s ich rezolventou. V prípade, že bolo potrebné nájsť pre nejaký rezolvenčný krok unifikátor, tak tento je v grafe znázornený tiež.

Okrem uvedených rezolvencií však mohli byť vykonané aj ďalšie. Takými mohli byť napríklad rezolvencia klauzuly $\neg susedne(S, Q) \vee \neg priepast(Q) \vee prievan(S)$ s klauzulou $\neg po(B, C) \vee susedne(pole(A, B), pole(A, C))$ či rezolvencia klauzuly $\neg susedne(S, Q) \vee \neg priepast(Q) \vee prievan(S)$ s klauzulou

Rozšírenie: Redukcia rovnosti

Binárny predikát < term > = < term > (vyjadrovaný častejšie v infixnom než v tradičnom prefixnom tvare) reprezentujúci rovnosť dvoch termov má oproti ostatným predikátom význačné postavenie. Môže byť síce eliminovaný pomocou rezolvencie ako každý iný predikát, avšak vďaka jeho sémantike sú aj iné spôsoby jeho eliminácie. Reprezentantom týchto iných spôsobov je *paramodulácia* založená na použití *paramodulačného pravidla*

PARAMODULÁCIA

$$\frac{P[t] \qquad (r = s) \vee Q}{subst(\theta, P[s] \vee Q\ )}$$

kde $P[t]$ je klauzula, ktorá obsahuje term t ako svoju súčasť (klauzula je kontextom pre tento term) a samotný term t nie je premennou). Potom θ je unifikátor, ktorý unifikuje term t a term r tak aby platilo

$$subst(\theta, t) = subst(\theta, r)$$

Intuitívne, ak $subst(\theta, Q)$ nie je pravdivá veta, potom musí platiť $subst(\theta, r) = subst(\theta, s)$. A teda $subst(\theta, t)$ môže byť nahradené nielen ekvivalentným $subst(\theta, r)$ ale aj odvodeným $subst(\theta, s)$. Príkladom paramodulácie je

$$\frac{q(X) \vee p(f(X)) \qquad (f(a) = g(a)) \vee q(b)}{q(a) \vee p(g(a)) \vee q(b)}$$

Paramodulácia sa často používa s cieľom nahradiť "väčšie" termy "menšími", teda je ohraničená na prípady, keď $subst(\theta, s)$ nie je podľa nejakého kritéria považované za väčšie ako $subst(\theta, r)$.

$\neg prievan(pole(1, 1))$. Obe tieto rezolvencie by vytvorili alternatívne cesty v derivačnom grafe k redukcii na prázdnu klauzulu (a v zmysle zobrazenej cesty dosiahnutia sporu by boli redundantnými).

Naproti tomu rezolvencia klauzuly $\neg susedne(S, Q) \vee \neg priepast(Q) \vee prievan(S)$ s klauzulou $\neg po(C, B) \vee susedne(pole(A, B), pole(A, C))$ vedie do slepej uličky, pretože pri aktuálnych znalostiach neumožní redukciu na prázdnu klauzulu (ostala by rezolventa $\neg po(2, 1)$, ktorú už nejde ďalej redukovať). Z hľadiska dosiahnutia prázdnej klauzuly by takáto rezolvencia bola zbytočná.

•

Rozšírenie: Zodpovedanie otázok

Výber informácií

Predpokladajme, že v ilustračnom príklade na strane 130 chlapci už pochodili časť priestoru a vykonali niektoré odvodenia (napr. že priepasť nie je na pozíciách $pole(1,3)$ a $pole(2,2)$ a že strašidlo sa nenachádza na pozíciách $pole(3,1)$ a $pole(2,2)$). Práve stoja na pozícii $pole(1,2)$ a potrebujú zistiť na základe svojej znalostnej bázy, ktoré pozície v okolí sú bezpečné. Otázka na vyhľadanie bezpečných pozícií by mohla mať tvar

$$\forall X\ (susedne(pole(1,2), X) \wedge \neg priepast(X) \wedge \neg strasidlo(X))$$

pričom táto otázka bude hrať úlohu hypotézy. Transformácia tejto hypotézy na tvar CNF a jej negácia poskytne cieľovú klauzulu

$$\neg susedne(pole(1,2), X) \vee priepast(X) \vee strasidlo(X)$$

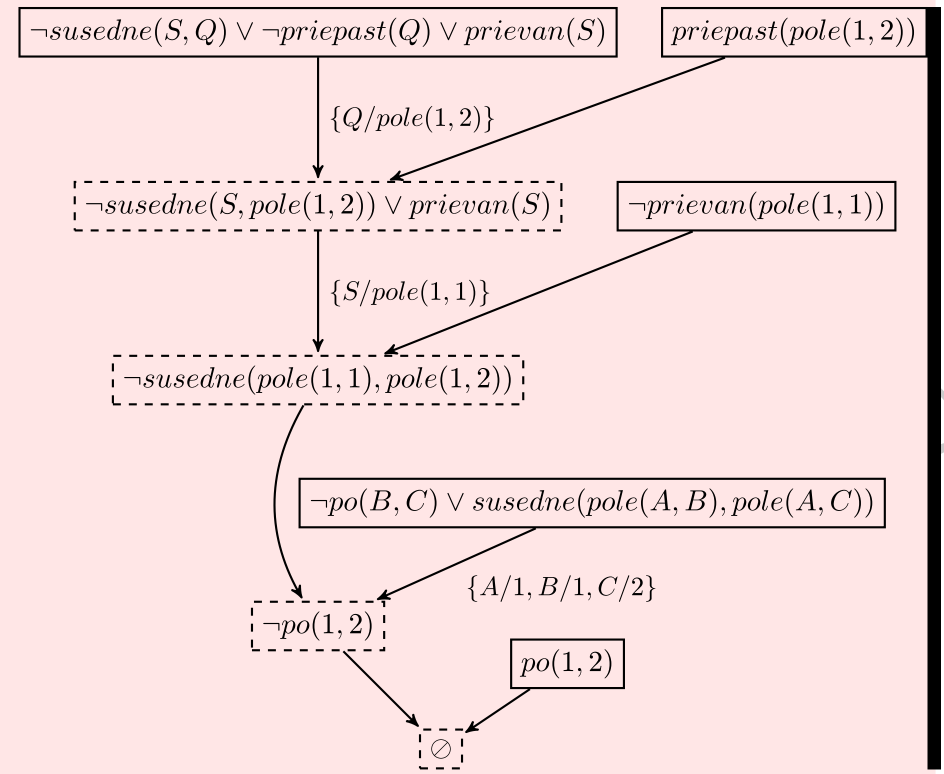
ktorú je možné pomocou série rezolvencií redukovať na prázdnu klauzulu použitím $\neg po(A,C) \vee susedne(pole(A,B), pole(C,B))$, $po(1,2)$, $\neg priepast(pole(2,2))$ a $\neg strasidlo(pole(2,2)$. Problémom je však to, že algoritmus Alg. 2.1 vráti $TRUE$ namiesto presného určenia bezpečnej pozície – teda podarilo sa dokázať, že bezpečná pozícia v okolí existuje, ale chlapci sa nedozvedeli, ktorá to je. Je síce možné namiesto premennej X použiť konkrétne označenie niektorého z polí (a riešiť osobitne pre každé pole), ale existuje aj iná možnosť.

V prípade, že cieľom nie je iba potvrdenie alebo vyvrátenie nejakej hypotézy ale výber informácií zo znalostnej bázy, možno k otázke doplniť ďalší "informačný" predikát, ktorý sa nenachádza v žiadnej klauzule v znalostnej báze. Argumenty tohto predikátu budú reprezentovať hľadanú informáciu. V našom prípade by potom po doplnení cieľová klauzula mala tvar

$$\neg susedne(pole(1,2), X) \vee priepast(X) \vee strasidlo(X) \vee info(X)$$

Keďže pre tento doplnený predikát v znalostnej báze neexistuje žiadny komplementárny predikát, nie je možná redukcia na prázdnu klauzulu, pretože nie je možnosť ho pomocou rezolvencie z klauzuly odstrániť.

Z tohto dôvodu algoritmus Alg. 2.1 musí byť upravený tak, aby namiesto testu na výskyt prázdnej klauzuly hľadal takú klauzulu, ktorá obsahuje iba doplnený informačný predikát (riadok 2 algoritmu) a v prípade úspechu namiesto $TRUE$ vrátil argumenty daného informačného predikátu (riadok 10). Aby to bolo možné, doplnený informačný predikát musí mať vopred definované meno, ktoré je algoritmu známe.



Obr. 5.2: Derivačný graf

Podobne ako pri výrokovej logike aj pri predikátovej logike je možné využiť tablo kalkul ako reprezentanta priamych metód, stavajúceho na využití pojmov splniteľnosti a validnosti. Rozdielom oproti výrokovej logike je to, že tam predstavený mechanizmus reprezentovaný prostredníctvom Alg. 2.2 je teraz potrebné modifikovať pre prácu s kvantifikátormi a nimu kvantifikovanými premennými. TABLO KALKUL

V zásade je potrebné používať dekompozičné pravidlá tablo kalkulu s dvomi typmi dodatočných pravidiel

- transformačnými pravidlami pre vety s výskytom kvantifikátorov,
- pravidlami pre odstránenie kvantifikátorov.

Príkladom transformačných pravidiel sú napríklad pravidlá pre distribučné vlastnosti kvantifikátorov. Pravidlá pre elimináciu kvantifikátorov sú zhrnuté v Tab. 5.6.

$\dfrac{\forall\ X\ p(X)}{p(t)}$	$\dfrac{\forall\ X\ \forall\ Y\ p(X,Y)}{p(t,t')}$	$\dfrac{\forall\ X\ \exists\ Y\ p(X,Y)}{p(t,f(t))}$
$\dfrac{\exists\ X\ p(X)}{p(a)}$	$\dfrac{\exists\ X\ \exists\ Y\ p(X,Y)}{p(a,a')}$	$\dfrac{\exists\ X\ \forall\ Y\ p(X,Y)}{p(a,t)}$

Tab. 5.6: Pravidlá odstránenia kvantifikátorov pre Tablo algoritmus

Odstraňovanie kvantifikátorov sa deje na základe konkretizácie, keď veta s kvantifikátorom je nahradená podobou bez kvantifikátora, pričom kvantifikovaná premenná je nahradená termom bez premennej (konštantou alebo funkciou bez premennej), ktorý reprezentuje niektorý objekt domény.

Napríklad odstránenie existenčného kvantifikátora je možné realizovať na základe

$$\exists\ X\ p(X)\ \models\ p(a)$$

kde a reprezentuje Skolemovu konštantu. Keďže $p(X)$ musí platiť aspoň pre jednu rozšírenú interpretáciu a teda aspoň jednu náhradu premennej za niektorý objekt domény, tak danú premennú za tento objekt domény možno nahradiť (ale keďže sa nevie, ktorý objekt by to mal byť, tak sa použije nové, doteraz nepoužité meno). Podobne odstránenie univerzálneho kvantifikátora je možné na základe

$$\forall\ X\ p(X)\ \models\ p(t)$$

kde t reprezentuje ľubovoľný objekt domény. Keďže teraz $p(X)$ musí platiť pre všetky rozšírené interpretácie a teda všetky náhrady premennej za niektorý objekt domény, tak potom bude pravdivé aj $p(t)$, keď sa premenná nahradí niektorým konkrétnym objektom (či sa už použije priamo meno objektu z domény alebo pomenovanie objektu skôr zavedené pri odstránení existenčného kvantifikátora).

Pri odstraňovaní viacerých kvantifikátorov každá z kvantifikovaných premenných sa nahrádza iným menom objektu, pričom je potrebné sledovať, či sa existenčný kvantifikátor nachádza v kontexte zovšeobecňovacieho kvantifikátora – ak áno, tak namiesto Skolemovej premennej je potrebné použiť Skolemovu funkciu. Tak napríklad $\exists\ X\ \forall\ Y\ \exists\ Z\ p(X,Y,Z)$ bude pomocou

$$\begin{aligned} \exists\ X\ \forall\ Y\ \exists\ Z\ p(X,Y,Z) &\models\ \exists\ X\ \forall\ Y\ p(X,Y,f(Y)) \\ &\models\ \exists\ Xp(X,t,f(t)) \\ &\models\ p(a,t,f(t)) \end{aligned}$$

nahradený $p(a, t, f(t))$, kde a je Skolemova konštanta, t je meno ľubovoľného objektu a $f(t)$ je Skolemova funkcia.

Príklad 5.9 Pre ilustráciu použitia Tablo kalkulu uvažujme ilustračný príklad na strane 130, pričom sa zamerajme iba na identifikáciu priepastí. Predpokladajme, že chlapci sa ocitli na poli $pole(1, 2)$, kde necítili žiadny prievan. Na základe toho sa pokúsia rozhodnúť, či na poli $pole(2, 2)$ je priepasť alebo nie.

Ak by $pole(2, 2)$ malo byť bezpečné bez priepasti, tak by muselo platiť $KB \models \neg priepast(pole(2, 2))$ kde KB je tento konjunktívny fragment znalostí chlapcov

$$
\begin{aligned}
& \forall\, S\ (prievan(S) \leftrightarrow \exists\, R\ (susedne(S, R) \wedge priepast(R))) \\
\wedge\ \ & \forall\, A, B, C\ (susedne(pole(A, B), pole(C, B)) \leftrightarrow\ po(A, C) \vee po(C, A)) \\
\wedge\ \ & po(1, 2) \\
\wedge\ \ & \neg prievan(pole(1, 2))
\end{aligned}
$$

kde sú obsiahnuté ako všeobecné znalosti o svete tak aj konkrétne znalosti získané pri skúmaní daného sveta.

Pre dokázanie platnosti daného vyplývania stačí dokázať validnosť vety $KB \rightarrow \neg priepast(pole(2, 2))$ alebo nesplniteľnosť jej negácie

$$F = \neg(KB \rightarrow \neg priepast(pole(2, 2))) = KB \wedge priepast(pole(2, 2))$$

Pre tento dôkaz je možné použiť Tablo kalkul – pre dokázanie nesplniteľnosti vytvorené tablo musí byť uzavreté a teda každá jeho vetva musí byť uzavretá (musí obsahovať spor). Výsledné tablo je na obr. 5.3. Jednotlivé formuly v table sú pre jednoduchšiu orientáciu indexované. Uzly v strome boli generované postupom:

- koreň stromu (1) obsahuje celú vetu, nesplniteľnosť ktorej je potrebné preskúmať,
- vety (2) až (6) vznikli z (1) pomocou pravidla pre dekompozíciu konjunkcie,
- vety (7) a (8) vznikli z (2) aplikáciou pravidla pre náhradu ekvivalencie pomocou dvoch implikácií

  $$\forall\, X\ (p(X) \leftrightarrow q(X))\ \equiv\ \forall\, X\ ((p(X) \rightarrow q(X)) \wedge (q(X) \rightarrow p(X)))$$

  s následným využitím pravidla distributívnosti univerzálneho kvantifikátora nad konjunkciou

  $$\forall\, X\ (r(X) \wedge s(X))\ \equiv\ \forall\, X\ r(X) \wedge \forall\, X\ s(X)$$

1: F

2: $\forall\ S\ (prievan(S) \leftrightarrow \exists\ R\ (susedne(S, R) \land priepast(R)))$

3: $\forall\ A, B, C\ (susedne(pole(A, B), pole(C, B)) \leftrightarrow\ po(A, C) \lor po(C, A))$

4: $po(1, 2)$

5: $\neg prievan(pole(1, 2))$

6: $priepast(pole(2, 2))$

7: $\forall\ S\ (prievan(S) \rightarrow \exists\ R\ (susedne(S, R) \land priepast(R)))$

8: $\forall\ S\ (\exists\ R\ (susedne(S, R) \land priepast(R)) \rightarrow\ prievan(S))$

9: $\forall\ S\ (\forall\ R\ (\neg susedne(S, R) \lor \neg priepast(R)) \lor\ prievan(S))$

10: $\neg susedne(pole(1, 2), pole(2, 2)) \lor \neg priepast(pole(2, 2)) \lor prievan(pole(1, 2))$

11: $\neg priepast(pole(2, 2))$
$\oslash$

12: $prievan(pole(1, 2))$
$\oslash$

13: $\neg susedne(pole(1, 2), pole(2, 2))$

14: $\forall\ A, B, C\ (susedne(pole(A, B), pole(C, B)) \rightarrow\ po(A, C) \lor po(C, A))$

15: $\forall\ A, B, C\ (po(A, C) \lor po(C, A) \rightarrow\ susedne(pole(A, B), pole(C, B)))$

16: $\forall\ A, B, C\ ((\neg po(A, C) \land \neg po(C, A)) \lor\ susedne(pole(A, B), pole(C, B)))$

17: $(\neg po(1, 2) \land \neg po(2, 1)) \lor\ susedne(pole(1, 2), pole(2, 2))$

18: $\neg po(1, 2) \land \neg po(2, 1)$

19: $susedne(pole(1, 2), pole(2, 2))$
$\oslash$

20: $\neg po(1, 2)$

21: $\neg po(2, 1)$
$\oslash$

Obr. 5.3: Splniteľnosť pomocou analytického tabla a záverečne aplikovaním pravidla pre dekompozíciu konjunkcie,

- veta (9) vznikla z (8) náhradou implikácie pomocou disjunkcie, pričom sa zohľadnila zmena typu kvantifikátora pri jeho negácii,
- veta (10) vznikla konkretizáciou (9), pričom boli použité objekty reprezentované funkciami $pole(1, 2)$ a $pole(2, 2)$,
- (11), (12) a (13) vznikli ako výsledok aplikácie pravidla pre dekompozíciu disjunkcie na vetu (10),
- vety (14) a (15) vznikli z (3) aplikáciou pravidla pre náhradu ekvivalencie pomocou dvoch implikácií s následným využitím pravidla distributívnosti univerzálneho kvantifikátora nad konjunkciou a záverečným použitím pravidla pre dekompozíciu konjunkcie,
- veta (16) vznikla z (15) náhradou implikácie pomocou disjunkcie,
- veta (17) vznikla konkretizáciou (16), pričom namiesto premenných A, B a C boli použité objekty 1, 2 a 2,
- (18) a (19) vznikli ako výsledok aplikácie pravidla pre dekompozíciu disjunkcie na vetu (17),
- (20) a (21) vznikli ako výsledok aplikácie pravidla pre dekompozíciu konjunkcie na vetu (18).

Vytvorené tablo má štyri vetvy, pričom všetky sú uzavreté – v prvej vetve je (11) v spore s (6), v druhej (12) je v spore s (5), v tretej (20) je v spore so (4) a v poslednej vetve (19) je v spore s (13). Znamená to, že skúmaná veta je nesplniteľná a teda na pozícii $pole(2, 2)$ sa nenachádza priepasť.

•

Tablo kalkul pri použití pre predikátovú logiku má jeden problém – ukončovanie otvorených vetiev. Príklad je na obr. 5.4, ilustrujúcom nevhodné poradie aplikovania pravidiel. NEKONEČNÝ ROZVOJ VETIEV

Najprv je premenná X nahradená konštantou a, reprezentujúcou objekt domény (2). Potom premenná Y je nahradená Skolemovou konštantou b (3). Keďže ešte vetva nie je uzavretá a objavila sa nová konštanta (konštanta b), tak (1) môže byť konkretizovaná použitím b pre premennú X (4). Následne je definovaná ďalšia nová Skolemova konštanta, čo umožní ďalšiu konkretizáciu (1) atď. Vetva môže byť takto rozvíjaná ešte hodne dlho.

Inou hrozbou je, ak doména obsahuje nekonečný počet objektov – a pred prehlásením vetvy za otvorenú je potrebné v rámci konkretizácie všeobecne

1: $\forall\ X\ \exists\ Y\ p(X,Y)$

2: $\exists\ Y\ p(a,Y)$

3: $p(a,b)$

4: $\exists\ Y\ p(b,Y)$

5: $p(b,c)$

6: $\exists\ Y\ p(c,Y)$

$\dots$

Obr. 5.4: Neukončená vetva analytického tabla

kvantifikovanej premennej preskúmať každú možnú konkretizáciu, či pri niektorej z nich predsa sa len neidentifikuje spor. Je potrebné si uvedomiť, že nekonečný počet objektov môže mať aj doména s jediným pomenovaným objektom a jednou unárnou reláciou, ktorú je možné nekonečne veľa krát reťaziť v tvare

$$otec(...otec(otec(jano))...)$$

čo nie je zriedkavým prípadom v reálnej doméne.

Slabinou metódy je teda rozvíjanie vetiev, ktoré sú v skutočnosti otvorené – toto môže viesť na nekonečnú expanziu. Z tohto dôvodu metóda budovania tabiel pri predikátovej logike nemôže byť považovaná za rozhodovaciu procedúru pre splniteľnosť v tejto logike. Preto sa pri predikátovej logike používa skôr pre dokazovanie validity viet cestou dokázania nesplniteľnosti negovanej podoby týchto viet.

Cvičenia

1. Reprezentujte nasledovné vyjadrenia v prirodzenom jazyku pomocou predikátového počtu prvého rádu a transformujte ich do tvaru CNF:

   - V meste je holič, ktorý holí všetkých mužov v meste, ktorí sa neholia sami.
   - Neexistuje také miesto, kde každý na tom mieste je chytrý vtedy a len vtedy, ak každý je na tom mieste a je chytrý.

- Všade je niekto, kto je tam a je chytrý ako aj je niekto, kto je chytrý ak tam je.
- Bratranci sú deti súrodencov.
- Všetci čitatelia si v každej knihe, ktorú čítali, našli svoju obľúbenú postavu práve vtedy, keď si v nej našli aj neobľúbenú postavu.

2. Unifikujte nasledovné dvojice výrazov

   - $p(f(X,a), g(Y,Y), Z) \qquad p(f(g(a,b), Z), X, a)$
   - $p(X, X, Z) \qquad p(f(a,a), Y, Y)$
   - $p(X, f(Y,Z), b) \qquad p(g(a,Y), f(Z, g(a,X)), b)$
   - $p(a, Y, U) \qquad p(X, f(X,U), g(Z,b))$

3. Pre ilustračný príklad na strane 130 prepíšte znalosti o príklade do tvaru CNF. Simulujte pohyb agenta, jeho vnemy postupne pridávajte do znalostnej bázy a na základe nich odvádzajte nové znalosti o jednotlivých poliach.

4. Vyriešte nasledujúcu detektívnu záhadu jedného tragického konca

   - Každého, kto má rád všetky zvieratá, má niekto rád.
   - Kohokoľvek, kto zabil nejaké zviera, nemá nikto rád.
   - Jack má rád všetky zvieratá.
   - Alebo Jack alebo zvedavosť zabili mačku, ktorá sa volala Fúzik.

   Bola vinníkom zvedavosť ?

5. Pri úlohe farbenia grafov je zadaný graf, ktorého uzly je potrebné zafarbiť – každému vrcholu priradiť jednu farbu tak, aby žiadne dva vrcholy, ktoré sú spojené hranou, neboli zafarbené rovnakou farbou. Počet farieb je vopred daný.

   Dokážte, že pre zafarbenie grafu s piatimi uzlami stačia štyri farby v prípade, že graf nie je plne prepojený, avšak v prípade, že každý uzol je spojený s každým iným uzlom, štyri farby nestačia.

6. Reprezentácie predchádzajúcich príkladov prepíšte do požadovaného tvaru a ich výsledky overte použitím implementácie algoritmu rezolvenčného dokazovania.

7. Dokážte validitu nasledovných viet

   - $\forall X\ p(X) \ \wedge\ \forall X\ q(X) \ \rightarrow\ \forall X(p(X) \wedge q(X))$

- $\exists\ X\ (p(X) \rightarrow q(X))\ \leftrightarrow\ (\forall\ X\ p(X) \rightarrow \exists\ X\ q(X))$
- $(\exists\ X\ p(X) \rightarrow \forall\ X\ q(X))\ \rightarrow\ \forall\ X\ (p(X) \rightarrow q(X))$
- $\forall\ X\ (p(X) \vee q(X))\ \rightarrow\ (\forall\ X\ p(X) \vee \exists\ X\ q(X))$
- $\forall\ X\ (p(X) \rightarrow q(X))\ \rightarrow\ (\exists\ X\ p(X) \rightarrow \exists\ X\ q(X))$
- $\forall\ X\ (p(X) \rightarrow q(X))\ \rightarrow\ (\forall\ X\ p(X) \rightarrow \forall\ X\ q(X))$

8. Šesť umeleckých diel (C, D, E, F, G a H) má byť vystavených v troch miestnostiach (1, 2 a 3) galérie. Výstava musí splniť tieto podmienky:

   - Diela C a E by nemali byť vystavené v rovnakej miestnosti.
   - Diela D a G musia byť vystavené v rovnakej miestnosti.
   - Ak E a F sú vystavené v rovnakej miestnosti, tak v tej miestnosti už by nemalo byť žiadne ďalšie dielo.
   - Aspoň jedno dielo musí byť vystavené v každej z miestností, najviac však tri diela v jednej miestnosti.

   Ak dielo D je vystavené v miestnosti 3 a diela E a F sú vystavené v miestnosti 1, ktoré z nasledujúcich tvrdení sú pravdivé ?

   - Dielo C je vystavené v miestnosti 1 (iné alternatívy: G v miestnosti 2, H v 1).
   - Nie viac ako dve diela sú vystavené v miestnosti 3.
   - Diela F a H sú vystavené v rovnakej miestnosti (iná alternatíva: C a H sú spolu).
   - Tri diela sú vystavené v miestnosti 2.

9. Uvažujte hádanku "Bratov ani sestier nemám, ale otec toho muža je synom môjho otca". Napíšte pravidlá pre doménu rodinných vzťahov a použite ich pre zistenie, kto je tým mužom.

10. Predpokladajme, že na základe znalostnej bázy $\{\ \forall X\ (c(X) \rightarrow g(X)),\ \forall X\ (a(X) \rightarrow \neg g(X)),\ \exists X\ (a(X) \wedge b(X))\ \}$ je potrebné dokázať hypotézu $\exists X\ (b(X) \wedge \neg c(X) \wedge \neg g(X))$. Dokážte ju pomocou metódy tablo kalkulu a aj pomocou rezolučnej metódy, pričom pri rezolvencii použite pre výber klauzúl stratégiu:

    - hĺbkovej saturácie,
    - usporiadania podľa veľkosti,
    - jednotkovej preferencie,
    - opornej množiny,
    - lexikografického (abecedného) usporiadania klauzúl.

Porovnajte stratégie z hľadiska počtu generovaných rezolvent a veľkostí vytvorených derivačných grafov.

Literatúra

1. Brachman, R.J. – Levesque, H.J.: Knowledge Representation and Reasoning. Morgan Kaufmann, San Francisco, CA, 2004.
2. Coppin, B.: Artificial Intelligence Illuminated. ones and Bartlett Publishers, 2004.
3. Draženská, E. -– Myšková, H.: Matematická logika. Technická univerzita v Košiciach, 2014.
4. Gabbay, D.M.: Logic for Artificial Intelligence and Information Technology. College Publication, 2007.
5. Lifschitz, V. – Morgenstern, L. – Plaisted, D.: Knowledge Representation and Classical Logic. In: Handbook of Knowledge Representation, van Harmelen et al. (eds.), Elsevier, 2008, 3–88.
6. Kvasnička, V.: Logika a využitie sémantických tabiel pre inteligentné systémy. Univerzita sv. Cyrila a Metoda, Trnava, 2018.
7. Mordechai B.-A.: Mathematic Logic for Computer Science (3. vyd.). Springer, London, 2012.
8. Návrat, P. a kol.: Umelá inteligencia. Slovenská Technická Univerzita, Bratislava, 2002.
9. Neapolitan, R.E. – Jiang, X.: Artificial Intelligence (2nd ed.).. CRC Press, 2018.
10. Russel, S. – Norvig, P.: Artificial Intelligence: A Modern Approach (3. vydanie). Pearson, Upper Saddle River, NJ, 2010.
11. Tao, T.: An Epsilon of Room, II: Pages from Year Three of a Mathematical Blog. American Mathematical Society, 2011.