Prostředky umělé inteligence

Predikátová logika

Jazyk predikátové logiky

Syntaxe predikátové logiky je soubor pravidel pro popis objektů termy. Termy se skládají z proměnných, funkcí a konstant. Nejjednodušší vztahy mezi termy, kterým lze přiřadit ***pravdivostní hodnotu*** (***true***/***false***) vyjadřují atomické formule pomocí predikátů.

Predikát
- odpovídá relaci,
- podle počtu termů může být unární, binární, ternární.

Atomickou formuli tvoří predikát s volbou termů v argumentech.

Pozn.:
Výčet všech konstant, funkcí a predikátů se specifikuje jazyk konkrétní predikátové logiky.

Jazyk predikátové logiky

Formule predikátové logiky vznikají z atomických formulí pomocí logických spojek a kvantifikátorů. Za základní se obvykle berou logické spojky konjunkce $\wedge$, disjunkce $\vee$, implikace $\rightarrow$ a negace $\neg$.

- Každá atomická formule je formule.
- Jsou–li α, β formule, pak výrazy
  $(\alpha \wedge \beta)$, $(\alpha \vee \beta)$, $(\alpha \rightarrow \beta)$, $\neg\alpha$,
  jsou rovněž formule.
- Je–li α formule a X proměnná, výrazy
  $\forall X\alpha$, $\exists X\alpha$
  jsou rovněž formule.
- Jiné formule neexistují.

Jazyk predikátové logiky

- ***Konstanty*** obvykle reprezentují konkrétní objekty.
- Volba ***predikátů*** je dána vztahy, které chceme používat.
- ***Funkce*** se používají pro přiřazení objektů jiným objektům.

Pozn.:
- Umožňují formulovat (speciální) axiomy.
- Obvykle se definuje nějaký pevně zvolený systém speciálních axiomů, který vymezuje uvažované prostředí

Formální systém matematické logiky

Je představován
- Jazykem predikátové logiky
- Axiomy
- Odvozovacími pravidly

Term t a formule α jazyka predikátové logiky lze považovat za slova (řetězce) v abecedě tvořené symboly uvažovaného jazyka a značkami $\wedge, \vee, \rightarrow, \neg, \forall, \exists, (,)$.

Precedence operátorů
$\neg, \forall, \exists, \wedge, \vee, \rightarrow$

Formální systém matematické logiky

- Je–li term t tvaru $f(t_1, t_2, \ldots, t_n)$, pak ***podtermy*** jsou termy $t_1, t_2, \ldots, t_n$. Totéž platí pro všechny podtermy těchto termů.
- Formule β je podformulí, je-li β podřetězcem formule α, který je současně formulí.
- Daný výskyt proměnné X ve formuli α je ***vázaný***, jestliže je součástí nějaké podformule tvaru $\forall X\beta$ nebo $\exists X\beta$. ***Proměnná je vázaná*** ve formuli α, má–li v ní vázaný výskyt. Není–li daný výskyt proměnné X vázaný, je ***volný***. ***Proměnná je volná*** ve formuli α, má–li tam volný výskyt.
- Formule je ***otevřená***, neobsahuje–li žádnou vázanou proměnnou.
- Formule je ***uzavřená***, neobsahuje–li žádnou volnou proměnnou.

Formální systém matematické logiky

Formule s čistými proměnnými je formule, v níž není žádná proměnná současně volná i vázaná.

Všechny proměnné, které nejsou explicitně kvantifikovány ve formuli jsou volné proměnné, tj. lze za ně dosadit libovolný term. Formule vzniklá dosazením se nazývá ***instancí*** původní formule.

Normální tvary formulí

Ke každé formuli φ predikátové logiky lze nalézt ekvivalentní formuli φ' (mají stejné pravdivostní hodnoty ve stejných situacích) v konjunktivně–disjunktivním tvaru (KDF). Základním představitelem KDF vez kvantifikátorů je klauzule.

Klauzule

- je konečná disjunkce literálů (atomických formulí nebo jejich negací),
- všechny proměnné klauzule jsou považovány za volné.

Formální systém matematické logiky

Zvláštním případem klauzule je ***prázdná klauzule*** $[\ ]$ s pravdivostní hodnotou false.

Konjunktivně disjunktivní forma je konjunkcí klauzulí. Může obsahovat kvantifikátory.

Prenexní normální forma obsahuje kvantifikátory pouze na začátku formule. Formule, která z ní vznikne odtržením kvantifikátorů je ***bekvantifikátorové jádro***.

Ke každé formuli existuje její bezkvantifikátorové jádro v konjunktivní normální formě.

Klauzule lze považovat za univerzální vyjadřovací prostředek predikátové logiky formulí s univerzálními kvantifikátory. Existenční kvantifikátory se eliminují tzv. Skolemizací, získá se ***Skolemův variant*** původní funkce.

Formální systém matematické logiky

- Přímý důkaz = používá odvozovací pravidla a axiomy k získání odvozované formule.
- Nepřímý důkaz = důkaz, že speciální axiomy jsou ve sporu s negací dokazované formule.

Rezoluční metoda

Rezoluční metoda je metodou hledání sporu v konečné množině klauzulí (nepřímá metoda).

$$T \mapsto \alpha \quad \textit{právě tehdy, když} \quad T \cup \left\{\neg(\alpha)\right\} \quad \textit{je sporná}.$$

Rezoluce je úplná dokazovací metoda (tj. je schopna nalézt spor v každé sporné množině klauzulí), proto lze rezoluci považovat za univerzální metodu důkazu vhodnou pro hledání důkazu).

Rezoluční metoda

Pozn.:

- Pro libovolnou množinu formulí (teorii M) lze nalézt odpovídající reprezentaci ve formě množiny klauzulí.
- Pro důkaz, že v nějaké teorii M obsahující axiomy ve tvaru základních klauzulí platí klauzule $\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n$, stačí ukázat, že teorie $M' = M \cup \left\{\neg\left(\alpha_1 \vee \alpha_2 \vee \ldots \vee \alpha_n\right)\right\}$ je sporná. V klauzulárním tvaru teorie je M', $M' = M \cup \left\{\neg\alpha_1, \neg\alpha_2, \ldots, \alpha_n\right\}$,
- kde $\neg\left(\neg\left(\beta\right)\right)$ je pro libovolnou formuli β je nahrazeno formulí β.

Mechanismus rezoluční metody

Spočívá v odvozování rezolvent pomocí rezolučního pravidla vždy ze dvou rodičovských klauzulí a jejich přidávání do teorie v klauzulárním tvaru M'.

- Přitom je každá rezolventa je logickým důsledkem rodičovských klauzulí, a tedy i celé výchozí množiny klauzulí. Hledáme–li zamítnutí, máme v každém kroku k dispozici konečnou množinu klauzulí, která je sjednocením výchozí množiny a všech jejích dosud vytvořených rezolvent. Z této množiny lze obecně vytvořit ne jednu, ale konečně mnoho dalších rezolvent.
- Důkaz sporu je dokončen, podaří–li se odvodit prázdnou klauzuli, která je sama nesplnitelná a má pravdivostní hodnotu false. V tom případě mluvíme o zamítnutí výchozí množiny klauzulí. Délka důkazu podstatně závisí na tom, které z možných rezolvent budeme postupně vybírat. Hovoříme o volbě strategie.

Pozn.:

Pro stručnost je někdy výhodné klauzule zapisovat jako množiny literálů.

Robinsonova věta

Libovolná množina klauzulí je sporná právě tehdy, je–li z ní odvoditelná prázdná klauzule po konečném počtu použití obecného rezolučního pravidla.

- Pro libovolný (tedy nejen rezoluční) algoritmus, který zjišťuje, zda zvolená formule je důsledkem vstupní množiny klauzulí, neexistuje žádné časové omezení pro délku jeho práce.
- Pokud je nějaká formule dokazatelná, pak algoritmus skončí, ale obecně nelze předem určit kdy.

Pozn.:

Jde o důsledek slavné Gödelovy věty o neúplnosti.

Rezoluční metoda / Pravidlo základní rezoluce

- Výchozí množina klauzulí obsahuje jen základní klauzule (tj. klauzule bez proměnných).
- Může být utvořena např. z libovolných uzavřených instancí obecných klauzulí.

Pravidlo:

Jsou – li φ a ψ dvě základní klauzule a je – li α atomická formule bez proměnných, pak z dvojice klauzulí $\alpha \vee \varphi$ a $\neg\alpha \vee \psi$ lze odvodit rezolventu $\varphi \vee \psi$ formulí $\alpha \vee \varphi$ a $\neg\alpha \vee \psi$. Literály α a $\neg\alpha$, které umožnily vzájemně rezolvovat obě uvažované formule, se nazývají doplňkovými literály.

Pozn.:

- Vychází z pravidla „modus ponens“, tj. rezolventa libovolné (konečné) množiny klauzulí M je splněna v každém modelu teorie M.
- Pokud z výchozí množiny M odvodíme rezolucí prázdnou klauzuli, pak teorie M nemůže mít žádný model (je sporná).
- Je–li teorie M sporná, pak v ní lze pomocí základní rezoluce odvodit prázdnou klauzuli.

Unifikace

Pro zobecněnou rezoluci má zásadní význam nejobecnější unifikace pro zvolenou konečnou množinu výrazů $\{E_i\}_{i \in I}$ tj. substituce s za volné proměnné studovaných výrazů taková, že:

i) pro všechny výrazy E_i, E_j platí, že $E_i(s) = E_j(s)$,

ii) pro libovolnou substituci u, splňující rovněž podmínku i) platí, že existuje substituce v taková, že pro každý výraz $E_j \in \{E_i\}_{i \in I}$ platí $(E_j(s))(v) = E_j(u)$, čili $u = s.v$.

Zobecněná rezoluce

Používá pravidlo obecné rezoluce.

Uvažujme dvě klauzule φ a ψ takové, že nemají společné proměnné. Nechť existují neprázdné klauzule φ_1, ψ_1 a φ_2, ψ_2 takové, že je splněno:

- $\varphi_1 \vee \varphi_2$ je φ a $\psi_1 \vee \psi_2$ je ψ,
- existuje nejobecnější unifikace s literálů z φ_1 a negací z literálů ψ_1. Takové rodičovské klauzule φ a ψ lze rezolvovat, výsledkem je klauzule $\varphi_2(s) \vee \psi_2(s)$, kterou nazýváme výsledná klauzule.

Pozn.:

Tytéž dvě klauzule mohou být obecně rezolvovány více rozdílnými způsoby.

Zobecněná rezoluce / Příklad

Teorie M je představována dvěma formulemi A_1, A_2. Má být zjišteno, je – li formule C v teorii M dokazatelná.

Teorie M a formule C v klauzulárním tvaru

A_1:	"Někteří pacienti mají rádi jakéhokoliv doktora."
	$\exists X (P(X) \wedge \forall Y (D(Y) \to R(X,Y)))$
	$P(a_1)$ $\neg D(Y_1) \vee R(a_1,Y_1)$
A_2:	"Žádný pacient nemá rád mastičkáře."
	$\forall X (P(X) \to \forall Y (M(Y) \to \neg R(X,Y)))$
	$\neg P(X_2) \vee \neg M(Y_2) \vee \neg R(X_2,Y_2)$
C:	"Žádný doktor není mastičkář."
$\neg$C:	$\neg(\forall Y (D(Y) \to \neg M(Y)))$
	$D(a_2)$ $M(a_2)$

Zobecněná rezoluce / Příklad

Aplikace rezoluční metody

1.	A_1:	$P(a_1)$	
2.	A_1:	$\neg D(Y_1) \vee R(a_1,Y_1)$	
3.	A_2:	$\neg P(X_2) \vee \neg M(Y_2) \vee \neg R(X_2,Y_2)$	
4.	$\neg$C:	$D(a_2)$	
5.	$\neg$C:	$M(a_2)$	
6.	4.,2.	$R(a_1,a_2)$	a_2/Y_1
7.	6.,3.	$\neg P(a_1) \vee \neg M(a_2)$	a_1/X_2 a_2/Y_2
8.	7.,5.	$\neg P(a_1)$	
9.	8.,1.	$\square$	

Rezoluční zamítnutí

Strategie zamítnutí může být

- úplná: pokud je prázdná klauzule z výchozí množiny odvoditelná, vede na sestrojení prázdné klauzule;
- neúplná: nemusí vést na sestrojení prázdné klauzule, i když tato je odvoditelná.

Derivační graf

- rezolventa je spojena hranami s oběma svými rodiči,
- listy grafu jsou výchozí klauzule,
- kořen grafu je prázdná klauzule.

Rezoluční zamítnutí je podgraf derivačního grafu, jehož

- listy jsou výchozí klauzule,
- kořen je prázdná klauzule.

Řád rezolventy

- je 0 pro vstupní rezolventu
- je i, je-li aspoň jeden z rodičů řádu i – 1

Rezoluční zamítnutí / Příklad

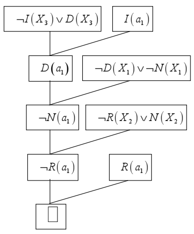
Teorie M a formule C v klauzulárním tvaru

A_1:	"Každý, kdo umí dokazovat věty, se nedá napálit."
	$\forall X (D(X) \rightarrow \neg N(X))$
	$\neg D(X_1) \vee \neg N(X_1)$
A_2:	"Roztržití lidé se dají napálit."
	$\forall X (R(X) \rightarrow N(X))$
	$\neg R(X_2) \vee N(X_2)$
A_3:	"Někteří roztržití lidé jsou inteligentní."
	$\exists X (R(X) \wedge I(X))$
	$R(a_1)$ $I(a_1)$
C:	"Jsou inteligentní lidé, kteří neumí dokazovat věty."
¬C:	$\neg(\exists X (I(X) \wedge \neg D(X)))$
	$\neg I(X_3) \vee D(X_3)$

Rezoluční zamítnutí / Příklad

Aplikace rezoluční metody

1.	A_1:	$\neg D(X_1) \vee \neg N(X_1)$	
2.	A_2:	$\neg R(X_2) \vee N(X_2)$	
3.	A_3:	$R(a_1)$	
4.	A_3:	$I(a_1)$	
5.	¬C:	$\neg I(X_3) \vee D(X_3)$	
6.	5.,4.	$D(a_1)$	a_1/X_3
7.	6.,1.	$\neg N(a_1)$	a_1/X_1
8.	7.,2.	$\neg R(a_1)$	a_1/X_2
9.	8.,3.	$\square$	



Rezoluční zamítnutí / strategie

Strategie prohledávání do šířky

postupně se generují všechny rezolventy i-tého řádu. Rezolventy $i+1$-tého řádu jsou generovány až tehdy, až jsou vygenerovány všechny rezolventy i-tého řádu.

Strategie podpůrné množiny

V každé výchozí množině klauzulí je možné určit podmnožinu, která je sama bezesporná. Rezolventy dvojic z této podmnožiny nemohou vést ke sporu.
Strategie podpůrné množiny generuje rezolventy jen z takových rodičovských párů, kdy jeden z rodičů je odvozen od negace dokazovaného tvrzení. Je tak buď přímo klauzulí, která vznikla při negaci dokazovaného tvrzení, nebo má takovou klauzuli za svého předchůdce.

Pozn.:

- Postupuje se do šířky.
- Důkaz může být delší, při prohledávání do šířky.
- Počet rezolvent všech řádů roste pomaleji u prohledávání do šířky.

Rezoluční zamítnutí / strategie

Jednotková strategie

generuje rezolventy z párů, ve kterých je alespoň jeden z rodičů klauzule tvořená jediným literálem.

Pozn.:

- Není úplná strategie.
- strategii lze chápat jako modifikaci podpůrné množiny, ve které se prohledávání do šířky zjemňuje jednotkovou preferencí.

Rezoluční zamítnutí / strategie

Vstupní strategie

generuje rezolventy z párů, ve kterých je alespoň jeden z rodičů přímo z výchozí množiny klauzulí.

Pozn.:

- Není úplná strategie.
- Jednoduchá a ve většině případů velmi efektivní.
- Existují různá rozšíření do úplné strategie.

Rezoluční zamítnutí / strategie

Filtrační strategie

Dvě klauzule α, β mohou být rodiči rezolventy, je–li

- α nebo β ze vstupní množiny nebo
- α ani β nejsou ze vstupní množiny, ale v derivačním grafu je buď α předchůdcem β, nebo β je předchůdcem α.

Pozn.:

- Jde o rozšíření vstupní strategie do úplné strategie.

Rezoluční zamítnutí / strategie

Lineární strategie

generuje rezolventy z párů, ve kterých je jeden z rodičů poslední generovaná klauzule.

Pozn.:

- strategií existuje několik
- nejde o úplné strategie
- použití v interaktivních systémech dokazování vět

Rezoluční zamítnutí / strategie

Pozn.:

- Některé z uvedených strategií je možné kombinovat.
- Časté spojení strategie podpůrné množiny se vstupní nebo filtrační strategií (každá z nich svým způsobem omezuje množinu rezolvent, ale neurčuje pořadí pro jejich generování).
- Většina strategií má řadu dalších zjemnění, které nějakým způsobem určují pořadí generovaných rezolvent. Nejjednodušším příkladem je jednotková preference (dává přednost rezolventám z jednotkových klauzulí). Další strategie využívají různých typů uspořádání literélů a klauzulí.

Omezování množiny rezolvent

Každá strategie prohledávání generuje i rezolventy, které z různých důvodů nemohou přispět k důkazu sporu. Některé jsou z nich snadno identifikovatelné. Tím, že se vynechají, se dále zmenší prohledávaný prostor.

Např.:

- Tautologie (obsahují doplňkový pár literálů) nemají žádný podíl na nesplnitelnosti množiny, lze je vypustit.
- Eliminace na základě logických hodnot literálů (až po substituci konstant) klauzulí, které obsahují literály s pravdivostní hodnotou true, zkracovat klauzule o literály, jejichž pravdivostní hodnota je false.
- Eliminace se může provést, je li jedna klauzule důsledkem jiné. říkáme, že klauzule γ_1 zahrnuje klauzuli γ_2, jestliže pro nějakou substituci s platí $\gamma_1(s) \subseteq \gamma_2$ (klauzule chápeme jako množiny literálů).

Omezování množiny rezolvent / Horn

Hornovy klauzule

jsou klauzule, které obsahují nejvýše jeden pozitivní literál.

Hornova logika

- formule transformovatelné na konjunkce Hornových klauzulí,
- vlatní podmnožina predikátové logiky,
- lze ji velmi efektivně zpracovávat,
- používá ji "logické programování" (PROLOG).

Umělá inteligence a logika

- Matematická logika je prostředek reprezentace znalostí.
- Zásadní princip budování logiky pro strojové dokazování je její ***monotónnost***:

$$\left(T \mapsto \varphi\right) \rightarrow \left(T^* \mapsto \varphi\right) \wedge \left(T^* \supseteq T\right)$$

Je–li φ dokazatelná v teorii T, pak je dokazatelná i v každé teorii $T^* \supseteq T$.

Pozn.:

- Monotónnost vyjadřuje předpoklad, že čím více bude výchozích axiómů, tím více nových tvrzení lze dokázat.
- V běžném usuzování může být monotónnost na překážku.

Příklad

Teorie $T=\{A_1,A_2,A_3\}$ ***a formule*** **C**

A_1:	„Každý pták umí létat."
	$\forall X(P(X)\rightarrow L(X))$
	$\neg P(X_1)\vee L(X_1)$
A_2:	"Tučňák je pták."
	$\forall X(T(X)\rightarrow P(X))$
	$\neg T(X_2)\vee P(X_2)$
A_3:	"Quido je tučňák."
	$T(Quido)$
C:	" Quido umí létat."
¬C:	$\neg(L(Quido))$

Aplikace rezoluční metody

1.	A_1:	$\neg P(X_1)\vee L(X_1)$	
2.	A_2:	$\neg T(X_2)\vee P(X_2)$	
3.	A_3:	$T(Quido)$	
4.	¬C:	$\neg L(Quido)$	
5.	1.,4.	$\neg P(Quido)$	$Quido/X_1$
6.	5.,2.	$\neg T(Quido)$	$Quido/X_2$
7.	6.,3.	$\sqcup$	

Tj. platí $T\mapsto C$.

Příklad

Teorie $T^*=\{A_1,A_2,A_3,A_4\}$ ***a formule*** $\neg C$

A_1:	"Každý pták umí létat."
	$\forall X(P(X)\rightarrow L(X))$
	$\neg P(X_1)\vee L(X_1)$
A_2:	"Tučňák je pták."
	$\forall X(T(X)\rightarrow P(X))$
	$\neg T(X_2)\vee P(X_2)$
A_3:	"Quido je tučňák."
	$T(Quido)$
A_4:	"Tučňáci nelétají. "
	$\forall X(T(X)\rightarrow \neg L(X))$
	$\neg T(X_3)\vee \neg L(X_3)$
¬C:	"Quido neumí létat."
C:	$\neg(\neg L(Quido))$

Aplikace rezoluční metody

1.	A_1:	$\neg P(X_1)\vee L(X_1)$	
2.	A_2:	$\neg T(X_2)\vee P(X_2)$	
3.	A_3:	$T(Quido)$	
4.	A_4:	$\neg T(X_3)\vee \neg L(X_3)$	
5.	¬C:	$L(Quido)$	
6.	5.,4.	$\neg T(Quido)$	$Quido/X_3$
8.	6.,3.	$\sqcup$	

Příklad

Tj. platí $T^*\mapsto\neg C$, ale současně, protože $T^*\supseteq T$, platí i $T^*\mapsto C$. Tím jsme zjistili, že teorie T^* obsahuje spor (její axiómy umožňují odvodit jak formuli, tak její negaci).

Neformální logiky

- Univerzální tvrzení v běžném životě mají řadu nevyslovených (implicitních) předpokladů. Jeden ze zdrojů nemonotónního usuzování jsou výjimky. Tím jsou inspirovány další formální systémy, mj. ***nemonotónní logika***.
- Jiným problémem je příliš „hrubá" rozlišovací schopnost pravdivostních hodnot true/false. Často se náš úsudek opírá o pravděpodobnostní hodnocení s daleko širší škálou hodnot (***pravděpodobnostní logika***). Se širší škálou hodnost pracuje také ***neostrá*** (***fuzzy***) ***logika***.
- Predikátová logika je vhodným prostředím uvažování pro určitý druh problémů. Pro jiné problémy se hledají a formalizují jiné modely. Spor o tom, zda je možné vybudovat inteligentní entitu pouze s použitím vhodného symbolického systému (***logicistický přístup***), či zda je nutné použití subsymbolických prvků jako jsou neuronové sítě (***konekcionistický přístup***).