Veštačka inteligencija

Logika predikata (predicate calculus)

Prof. dr Leonid Stoimenov

Logika predikata kao formalizam za predstavljanje znanja

Formalni jezik za predstavljanje znanja, definisan azbukom i simbolima **preko kojih se grade formule (rečenice).**

Ekstenzija logike iskaza (propositional logic)

Istinitost formula predikatske logike se utvrđuje pomoću **uzastopne primene relacija** iz istinitonosnih tablica, počev od **unutrašnjosti prema spoljašnjosti** formule.

Logike dugog reda: nemonotone logike, temporalne logike, kvalitativno rezonovanje, belief management, planiranje i prepoznavanje planova, …

Predosti logike predikata prvog reda:

- Postoji matematički aparat za zaključivanje
- Postoji preslikavanje na i sa prirodnog jezika

Nedostaci:

- Teškoće u predstavljanju aproksimativnog, heuristickog zdravorazumskog znanja, znanja o planovima, verovanja ili znanja o vremenu

Elementi jezika

Azbuka – čine je specijalni znaci:

. , ( ) { } [ ] + - * / ^ ∈ ∩ ∪ ⊂ ⊃ ⊆ ⊇

= < > <= >= ¬ ∧ ∨ ∀ ∃ => <= ⇔

Simboli:

- **Konstante**: slova i cifre, prvo slovo veliko (A, B, X45, Ivan, …); za ime konstante se koriste slova i cifre, ali ako je slovo prvo onda ono mora biti veliko
- **Promenljive**: mala slova za ime, prvo slovo je veliko, cifre (x, y)
- **Funkcije**: prvo slovo veliko, cifre (Sinus, Otac, …)
- **Predikati** (relacije): prvo slovo veliko (Veci-od, Brat, …), odgovaraju matematičkim relacijama
- **Logicki konektori** (operacije): $\neg$, $\wedge$, $\vee$, $\Rightarrow$, $\Leftrightarrow$
- **Kvantifikatori**: $\forall$, $\exists$

Izrazi: referenciraju objekte u stvarnom svetu: konstante, promenljive i funkcije

Logika predikata prvog reda

Najčešće korišćeni formalni logički sistem

Predikati prvog reda **<u>ne dozvoljavaju</u> kvantifikaciju nad predikatima i funkcijama**.

Pretpostavka da svet sadrži:

- **Objekte**: ljudi, kuće, brojevi, boje, igre loptom, ratovi, …
- **Relacije** između objekata: crven, okrugao, prost broj, brat, veći od, deo, između, …

Prevodjenje iz prirodnog jezika u logiku predikata

Postupak izgradnje znanja pomoću formalne logike se sastoji od sledećih faza:

1. Formulacija znanja u obliku rečenica govornog jezika,
2. Razdvajanje rečenica (iskaza) na njihove komponente,
3. Izbor simbola koji će predstavljati elemente i odnose u svakom iskazu i
4. Izgradnja WFF (korišćenjem definisanih simbola) koje će predstavljati iskaze.

Primer

1) **Poznato je:**

   Sve mačke imaju rep.

   Tom je mačka.

Doneti zaključak:

   Tom ima rep.

2) **Poznato je:**

   Ako je dan sunčan, sunce sija na prozoru.

   Ako sunce sija na prozoru, roletne su spuštene.

   Roletne nisu spuštene.

Doneti zaključak:

   Da li je sunčano danas?

Prevodjenje iz prirodnog jezika u logiku predikata

Relacije - Predikati

- Neke relacije su **osobine**: odnose se na neke činjenice o **jednom** objektu:

  Okrugla(lopta) ili Prost(7)

- n-arne relacije se odnose na **činjenice** o dva ili više objekata:

  Oženjen(Jovan,Marija), Veći_od(3,2).

- Neke relacije su **funkcije**: njihova vrednost je drugi objekat:

  plus(2,3), otac(Dane).

Razlike: Funkcija: otac(Marija) = Bojan
Predikat: Otac(Marija, Bojan)

Svaki simbol ima **interpretaciju**:

"Jovan" se odnosi na osobu Jovan.
"Brat" se odnosi na relaciju brat.
"Otac" se odnosi na relaciju odnosno predikat

Dobro-formirane formule (Well-Formed Formulas, WFF, rečenice)

Induktivno se definišu kao:

1. Atomske formule (predikati)

$\rho(\tau 1, \tau 2, \ldots, \tau n)$

gde je ρ n-arna relacija (predikat), a $\tau 1, \tau 2, \ldots, \tau n$ su termi (konstante, promenljive, funkcije)

Primer 1:	Primer 2:
Živeti (Pera, Kuća1)	Nosi(A, B)
Kupiti (Otac(Mika), Kuća1)	Kocka(A)
Vece(Abs(-5), 3)	Lopta(B)
Otac(Dejan, Ana)	Sto(C)
Boja(x, Plava)	Lezi_na(B, A)
Jednakost: Otac(Mika)=Petar	

2. **Logičke formule:**

atomske formule + logicki operatori ($\neg$ $\wedge$ $\vee$ => <= $\Leftrightarrow$)

Primeri logičkih formula:

Pas (Lesi) => Životinja (Lesi) ← Ako je Lesi pas onda je i životinja.

Broj(x) $\wedge$ Vece(x, y)

Slon(Dambo) $\Rightarrow$ Leti(Dambo)

$\neg$Mladji(Nikola, Violeta)

3. **Kvantifikovane formule:** logicke formule + kvantifikatori $\forall$, $\exists$

Primeri kvantifikovanih rečenica:

$\forall$x Slon(x) $\Rightarrow$ Boja(x, Siva)

$\forall$x Ptica(y) $\Rightarrow$ Leti(y)

$\exists$z Ptica(z) $\wedge$ $\neg$Leti(z)

$\forall$x$\forall$y Nosi(x, y) => Iznad(y, x)

$\forall$x$\forall$y$\forall$z Iznad(x, y) $\wedge$ Iznad(y, z) => Iznad(x, z)

Primer

1) Predikati

Mačka(x) – x je mačka

ImaRep(x) – x ima rep

Sve mačke imaju rep.	$\forall x$ Mačka(x) $\Rightarrow$ ImaRep(x)
Tom je mačka.	Mačka(Tom)

Doneti zaključak:

Tom ima rep.	ImaRep(Tom)

2) Poznato je:

Ako je dan sunčan, sunce sija na prozoru.

Ako sunce sija na prozoru, roletne su spuštene.

Roletne nisu spuštene.

Doneti zaključak:

Da li je sunčano danas?

Kvantifikacija

Za izražavanje tvrdnji "**za svako** …" i "**postoji** …"

- Univerzalni kvantifikator, $\forall$
- Egzistencijalni kvanitifikator, $\exists$

Primer:

- Sve mačke su životinje.

$$\forall x \; Macka(x) \Rightarrow Zivotinja(x)$$

- Postoji mačka čiji je vlasnik Pera.

$$\exists x \; Macka(x) \wedge Vlasnik(x, Pera)$$

Oblast delovanja kvantifikatora

Redosled kvantifikatora je važan !!

Ugnježdenje kvantifikatora – kombinacija univerzalnog i egzistencijalnog u istoj rečenici (oblast delovanja je formula uklopljena u kvantifikovanu rečenicu).

Primer: Uporediti značenje sledećih rečenica

$\forall x\ \exists y\ Voli(x,y)$

$\exists y\ \forall x\ Voli(x,y)$

Tumačenja, u zavisnosti od redosleda kvantifikatora:

$((\forall x) ((\exists y) Voli(x,y))) \leftarrow$ Za svako x postoji y za koje x voli y

$((\exists y) ((\forall x) Voli(x,y))) \leftarrow$ Postoji neko koga svi vole

Veze izmedju kvantifikatora:

$\forall x \neg P \equiv \neg \exists x P$

$\forall x\ P \equiv \neg \exists x \neg P$

$\neg \forall x\ P \equiv \exists x \neg P$

$\exists x P \equiv \neg \forall x \neg P$

Dodatne definicije termina:

Slobodna promenljiva:

Ako se promenljiva javlja u izrazu **bez kvantifikatora** onda je to slobodna promenljiva.

$Jabuka(x) \Rightarrow Crvena(x)$ ← *x* je slobodna promenljiva

$(\forall x)\ (\ Jabuka(x) => Crvena(x)\ )$ ← x uz $\forall$ je vezana, a u ostala dva slučaja slobodna promenljiva

$(\ Jabuka(x)\ \vee\ (\ (\exists x)\ Crvena(x)\ )\ )$ ← x u Jabuka(x) je slobodna, a desno od simbola $\vee$ x je vezana promenljiva

Zatvorena rečenica: ona rečenica u kojoj nema slobodnih promenljivih.

Interpretacija je preslikavanje elemenata jezika na objekte iz sveta.

Valjane formula je ona i samo ona formula koja je tačna za svaku moguću interpretaciju vrednosti promenljivih, funkcija i predikata.

Tautologija je valjana formula bez promenljivih: $((P \Rightarrow Q) \Rightarrow P) \Rightarrow P$

*Logika predikata **prvog reda**:* kvantifikatori se primenjuju na promenljive.

Logika predikata drugog reda: kvantifikatori se primenjuju na funkcije i predikate

Primeri (kvantifikatori) – razmotrite rečenice:

1) Majka osobe je i roditelj te osobe.
   Predikati:
   Osoba(x) – x je osoba
   Roditelj(x) – x je roditelj
   Funkcija: majka-od(x)

   - $\forall$x Osoba (x) $\Rightarrow$ Roditelj(majka-od(x),x)

   vs.

   - $\forall$x Osoba (x) $\wedge$ Roditelj(majka-od(x),x)

2) Postoje osobe koje misle da je ovaj kurs cool.
   Predikati:
   VI-je-cool(x) – x misli da je kurs VI cool

   - $\exists$x Osoba (x) $\wedge$ T (x)

   vs.

   - $\exists$x Osoba (x) $\Rightarrow$ T (x)

"Preporuke":

- $\Rightarrow$ obično ide sa $\forall$, $\wedge$ obično ide sa $\exists$

$$\forall x, King(x) \Rightarrow Person(x) \quad x = \{Pete, Mary, tablespoon\}$$

$$\forall x, King(x) \wedge Person(x)$$

$$\exists x, King(x) \Rightarrow Person(x)$$

$$\exists x, King(x) \wedge Person(x)$$

Za sve mora da bude tačno!
King(Pete) AND Person(Pete)
King(Mary) AND Person(Mary)
King(Tablespoon) AND Person(Tablespoon)
Isuviše jako!

Jedno bi trebalo da bude tačno!
if King(Pete) then Person(Pete)
if King(Mary) then Person(Mary)
If King(Tablespoon) then Person(Tablespoon)
Isuviše slabo!

Restrikcija kvantifikatorima

Predstaviti "Svi studenti ___ ."

$(\forall x \;\; Student(x) \Rightarrow \_\_\_\_\_)$

Predstaviti "Student ___." "Neki studenti ___."

$(\exists x \; Student(x) \wedge \_\_\_\_\_)$

Predstaviti "Nema studenata koji ___."

$(\forall x \;\; Student(x) \Rightarrow \neg \_\_\_\_\_)$

Svi studenti ne ___.

$\neg (\exists x \; Student(x) \wedge \_\_\_\_\_)$

Postoje studenti koji ___. $\exists x \; Student(x) \wedge \_\_\_\_\_$

Negacija i kvantifikatori

Promena kvantifikatora kod “provlačenja” negacije

“Svi advokati nisu naivni” *je ekvivalentno sa* “Ne postoji advokat koji je naivan”

$$\forall x\ Advokat(x) \Rightarrow \neg Naivan(x)$$

$$\forall x\ \neg Advokat(x) \vee \neg Naivan(x)$$

$$\forall x\ \neg(Advokat(x) \wedge Naivan(x))$$

$$\neg\exists x\ Advokat(x) \wedge Naivan(x)$$

Ugnježdeni kvantifikatori

Povezati!

$\exists x \exists y\ \ Macka(x) \wedge Vlasnik(x,y)$	*Za sve vlasnike važi da vlasnik voli ono što poseduje.*
$\forall x \exists y\ \ Macka(x) \Rightarrow Vlasnik(x,y)$	*Postoji neko ko je vlasnik svih mačaka*
$\exists y \forall x\ \ Macka(x) \Rightarrow Vlasnik(x,y)$	*Sve mačke imaju vlasnika (ali vlasnik može biti različit!).*
$\forall x \forall y\ \ Vlasnik(x,y) \Rightarrow Voli(x,y)$	*Postoji mačka koja ima svog vlasnika.*

Primeri prevođenja iz prirodnog jezika u WFF

Švrća je mali pas.

Pas (Švrća) ∧ Mali (Švrća)

Pera je student, ali nije ni pilot, ni fudbaler.

Student (Pera) ∧ ⌉Pilot (Pera) ∧ ⌉Fudbaler (Pera)

Pera voli da jede sve (bilo koju vrstu hrane).

∀x Hrana(x) => Voli (Pera,x)

Pera voli bar jedno jelo koje Mica voli.

∃x Hrana(x) ∧ Voli(Mica, x) ∧ Voli(Pera, x)

Svaki čovek koji ne vežba brzo se ugoji.

(∀ x)(Čovek (x) ∧ ⌉Vežbati (x) ⇒ Brzo_se_ugoji (x))

Svako Perino jelo je začinjeno.

∀x Jelo(Pera,x) => Začinjeno(x)

Svako voli neku hranu. (važan je redosled kvantifikatora!!)

∀x ∃y Hrana(y) ∧ Voli(x,y)

Postoji hrana koju svi vole. (važan je redosled kvantifikatora!!)

∃y ∀x Hrana(y) ∧ Voli(x,y)

Kad god neko jede začinjena jela, srećan je. (važan je redosled kvantifikatora!!)

$\forall$x $\exists$y Hrana(y) $\wedge$ Začinjeno(y) $\wedge$ Jede(x,y) => Srećan(x)

Svaki grad ima šintera, koji je bio ujeden od svakog psa u gradu.

$\forall$t $\exists$c $\forall$d Grad(c) => (Šinter (c,t) $\wedge$
(Pas(d) $\wedge$ Živi-u (d, t) $\rightarrow$ Ujeo (d, c)))

Za svaki skup X postoji skup Y takav da je kardinalni broj skupa Y veći od kardinalnog broja skupa X.

Predikati: Kard(X, u), Skup(X), Veće(X, Y)
$\forall$X Skup(X) => $\exists$Y $\exists$u $\exists$v Skup(Y) $\wedge$ Kard(X, u) $\wedge$ Kard (Y, v) $\wedge$ Veće(v, u)

Sve crvene pečurke su otrovne.

$\forall$x Pecurka(x) $\wedge$ Crvena(x) => Otrovna(x)
$\forall$x Pecurka(x) => (Crvena(x) => Otrovna(x))

Pečurka je otrovna jedino ako je crvena.

$\forall$x Pecurka(x) $\wedge$ Otrovna(x) => Crvena(x)
$\forall$x Pecurka(x) => (Otrovna(x) => Crvena(x))

Nijedna crvena pečurka nije otrovna.

$\forall$x $\neg$(Pecurka(x) $\wedge$Crvena(x) $\wedge$ Otrovna(x))
$\neg$($\exists$x) (Pecurka(x) $\wedge$Crvena(x) $\wedge$ Otrovna(x))

Primeri za vežbu

Ivan drzi plavi dijamant u ruci.
Ako je x iznad y, onda je y ispod x.
Svaka zemlja ima glavni grad.
Ne postoji broj veci od 5.
Nije tacno da svaki zidar pusi.
Kaput u Marijnom ormanu pripada Nikoli.
Zoran je uradio nesto sto je naljutilo Anu.
Svi ljudi u ovoj sobi govore isti jezik.
Mozda cu doci na zabavu, a mozda necu.
Samo jedan student je pao na ispitu iz VI.

Rešenja

$\text{Dijamant}(x) \wedge \text{Plavo}(x) \wedge \text{Drži}(\text{Ivan},x)$

$\forall x\ \forall y\ \text{Iznad}(x,y) \Rightarrow \text{Ispod}(z,x)$

$\forall x\ \exists y\ (\text{Zemlja}(x) \wedge \text{Grad}(y)) \Rightarrow \text{GlavniGrad}(y,x)$

$\forall x\ \text{Broj}(x) \Rightarrow \neg\text{Veći}(x,5)$

$\exists x\ \text{Zidar}(x) \wedge \neg\text{Puši}(x)$

$\exists x\ \text{Kaput}(x) \wedge \text{Pripada}(x, \text{Nikola}) \wedge \text{U}(x, \text{Orman}(\text{Marija}))$

$\exists x\ \text{Uradio}(\text{Zoran},x) \wedge \text{Naljutio}(\text{Ana}, x)$

$\forall x\ (\text{čovek}(x) \wedge \text{U}(x, \text{OvaSoba})) \Rightarrow \exists y\ \text{Jezik}(y) \wedge \text{Govori}(x, y)$

$\text{Dolazim}(\text{Ja}) \vee \neg\text{Dolazim}(\text{Ja})$

$\exists x\ \text{Student}(x) \wedge \neg\text{Položio}(x, \text{VI}) \wedge \forall y\ \text{Student}(y) \wedge \neg\text{IstaOsoba}(x, y) \Rightarrow \text{Položio}(y, \text{VI})$

Poklapanje uzoraka

Postupak u kome se ispituje da li **neka formula može da postane istoventa sa datim podatkom**, ako se promenljive zamene odredjenim vrednostima.
Podatak je iskaz o stvarnom ili zamišljenom svetu (formula bez promenljivih).

Lista smena je skup asocijacija izmedju promenljivih i izraza u kojima je svaka promenljiva vezana za najvise jedan izraz.

Lista smena – skup parova promenljiva/vrednost. {x/A, y/F(B), z/C}

Primeri:

Otac (Nikola, Petar)	P(A, B, x, D, E)	Q(A, B, x, D)
Otac(x, Petar)	P(A, B, Q(C), D, E)	Q(A, C, B, D)
Lista smena {x/Nikola}	Lista smena: ??	Lista smena: ??

Ograničenja:

- Ako uzorak (formula) ne sadrzi promenljive, ona se poklapa sa podatkom jedino ako su identični.
- Kada se promenljiva pojavljuje više puta u uzorku, onda se ona uzima kao nepoznata samo prvi put, a kasnija pojavljivanja su vezana za prvo.

Lista smena (binding list) se moze **primeniti na formulu**:

P(x, x, y, z) {x/A, y/F(B), z/C} = P(A, A, F(B), C)

Unifikacija

Unifikacija (objedinjavanje) je proces u kome se utvrdjuje **da li neka formula moze da bude identicna sa drugom formulom** (generalizovano poklapanje uzoraka).

Unify(p,q)=θ, tako da subst(θ,p)=subst(θ,q)

Dodatni uslov za listu smena: nijedna promenljiva sa dodeljenim izrazom ne javlja se u asociranom izrazu, ni u izrazima dobijenih iz njega zamenom promenljivih.

{x/A, y/v, z/w} ??
{x/G(y), y/F(x)} ??
{x/y, y/z, z/x} ??

Primeri poklapanja uzoraka i unifikacije

P(A,x) P(A,B)

Lista smena: x/B Primena liste smena: P(A,B)

P(A,x) P(y,B)

Lista smena: y/A, x/B P(A,B)

P(A,x) P(y, f(y))

Lista smena: y/A, x/f(A) P(A,f(A))

P(A,x) P(x,B)

failure

P(A,x) P(y,B)

Lista smena:: y/A, x/B P(A,B)

P(A,B) P(x,x)

failure

P(x, f(y), B) P(x, f(B), B)

Lista smena: y/B

(Praktikum):

Algoritam: Unify(p, q, s)

Ako su **p** i **q** jednaki, izlaz **s**.
Ako su **p** i **q** razliciti atomi, izlaz **nil**.
Ako je jedan od uzoraka promenljiva, zovi *unify-var*.
U suprotnom, **p** i **q** moraju biti liste.
Ako su liste razlicite duzine, izlaz **nil**.
Zovi *unify-var* za odgovarajuce komponente **p** i **q**.
Svaki poziv ove funkcije vraća **nil** ili novu listu smena.
Ako je rezultat **nil**, izlaz **nil**.
U suprotnom, zameni **s** novom listom smena.
Ako su sve komponente uzoraka obradjene, izlaz **s**.

Algoritam: unify-var(var, pat, s)

Ako **var** ima dodeljenu vrednost u **s**, vrati rezultat unifikovanja te vrednosti sa **pat**.
Ako je **var=pat**, izlaz **s**.
U ostalim slucajema, proveri da li se **var** javlja u **pat**.
Ako da, izlaz **nil**.
Ako ne, dodaj novu smenu **var/pat** listi **s** i vrati **s**.

Zaključivanje u logici predikata prvog reda

Izvodjenje (inference) je process dobijanja zaključaka iz premisa.

Opšta pravila izvodjenja se primenjuju se na formule da bi se dobile nove formule. Svako pravilo se sastoji od uslova i zaključaka. Kada su prisutne formule koje odgovaraju uslovima, mogu se izvesti formule koje odgovaraju zaključcima.

Primena pravila izvođenja

To je **proces dobijanja zaključaka** iz premisa. Pravila izvođenja se primenjuju na formule da bi se dobile nove formule. Svako pravilo čine uslovi i zaključci.

Zaključak Φ je izvodljiv iz skupa premisa Δ ako i samo ako:

1. Φ je član Δ
2. Φ je rezultat primene pravila izvođenja na formule izvodljive iz Δ

Pravila zaključivanja su:

1) MP (MODUS PONENS)

$\Phi \Rightarrow \Psi$ ← premisa

Φ ← premisa

Ψ ← zaključak

Nosi(A, B)

Nosi(A, B) => Iznad(B, A)

Iznad(B, A)

2) MT (MODUS TOLENS)

$\Phi \Rightarrow \Psi$

$\neg \Psi$

$\neg \Phi$

3) AND ELIMINATION (AE) Eliminacija knjunkcije

$\Phi \wedge \Psi$

Φ

Ψ

4) AND INTRODUCTION (AI) Uvođenje konjunkcije

Φ

Ψ

$\Phi \wedge \Psi$

5) UNIVERSAL INSTANTIATION (UI) Eliminisanje univerzalnog kvantifikatora

$\forall \; \nu \; \Phi$

$\Phi_{\nu/\tau}$ ← τ menja ν

τ je izraz u kome se ν ne javlja kao slobodna promenljiva

$\forall$y Voli(Jelena, y)

Voli(Jelena, Marko)

6) EXISTENTIAL INSTANTIATION (EI) Eliminisanje egzistencijalnog kvantifikatora

$\exists \; \nu \; \Phi$

$\Phi \; \nu / \pi(\nu 1, \nu 2, \ldots, \nu n)$

π je funkcija, a $\nu 1$, $\nu 2$, …, νn su slobodne promenljive u funkciji Φ

$\exists$z Mrzi(y, z)

Mrzi(y, f(y))

Primer zaključivanja:

P

P => Q

P => R ← Δ

Q => S

Q

R

S

Primer:

Po zakonu je kažnjivo prodavanje neregistrovanih pištolja. Pera ima nekoliko neregistrovanih pištolja koje je kupio od Sime. Izvesti zaključak da je Sima kriminalac.

Rešenje:

P(x, y, z) ← x je prodao y osobi z, y je pištolj
N(y) ← y je neregistrovan pištolj
I(x, y) ← x ima y
K(x) ← x je kriminalac

1) $\forall x \forall y \forall z \quad P(x, y, z) \wedge N(y) \Rightarrow K(x)$ … Δ

2) $\exists y\ I(Pera, y) \wedge N(y)$ … Δ

3) $\forall y\ I(Pera, y) \wedge N(y) \Rightarrow P(Sima, y, Pera)$ … Δ

4) $I(Pera, P1) \wedge N(P1)$ … iz 2) EI

5) $I(Pera, P1) \wedge N(P1) \Rightarrow P(Sima, P1, Pera)$ … iz 3) UI

6) $P(Sima, P1, Pera)$ … iz 4) i 5) MP

7) $I(Pera, P1)$ … iz 4) AE

8) N(P1) … iz 4) AE

9) P(Sima, P1, Pera) $\wedge$ N(P1) … iz 6) i 8) AI

10) P(Sima, P1, Pera) $\wedge$ N(P1) => K(Sima) … iz 1) UI

11) K(Sima)

Primer:
Konji su brži od pasa, a lovački psi od zečeva. Duško je zec, a Munja je konj. Izvesti zaključak da je Munja brži od Duška.

Rešenje:

Konj(x) ← x je konj
Zec(x) ← x je zec
Pas(x) ← x je pas
LovačkiPas(x) ← x je lovački pas
Brži(x, y) ← x je brži od y

1) $\forall x \forall y$ Konj(x) $\wedge$ Pas(y) => Brži(x, y) … Δ

2) $\exists y$ LovačkiPas(y) $\wedge$ ($\forall z$ Zec(z) => Brži(y, z)) … Δ

// $\exists y$ zbog lakšeg rešenja

3) $\forall y$ LovačkiPas(y) => Pas(y) … Δ

4) $\forall x \forall y \forall z$ Brži(x, y) $\wedge$ Brži(y, z) => Brži(x,z) … Δ

5) Konj(Munja) … Δ

6) Zec(Duško) … Δ

7) LovačkiPas(Žuća) $\wedge$($\forall z$ Zec(z) => Brži(Žuća,z)) … iz 2) EI

8) LovačkiPas(Žuća) … iz 7) AE

9) $\forall z$ Zec(z) => Brži(Žuća, z) … iz 7) AE

10) Zec(Duško) => Brži(Žuća, Duško) … iz 9) UI

11) Brži(Žuća, Duško) … iz 10) i 6) MP

12) LovačkiPas(Žuća) => Pas(Žuća) … iz 3) UI

13) Pas(Žuća) … iz 12) i 8) MP

14) Konj(Munja) $\wedge$ Pas(Žuća) => Brži(Munja, Žuća) … iz 1) UI

15) Konj(Munja) $\wedge$ Pas(Žuća) … iz 5) i 13) AI

16) Brži(Munja, Žuća) … iz 14) i 15) MP

17) Brži(Munja, Žuća) $\wedge$ Brži(Žuća, Duško) => Brži(Munja, Duško) … iz 4) UI

18) Brži(Munja, Žuća) $\wedge$ Brži(Žuća, Duško) … iz 11) i 16) AI

19) Brži(Munja, Duško) … iz 17) i 18) MP

Rezolucija

Rezolucija je procedura izvođenja koja se zasniva na pravilu izvođenja koje je poznato kao **princip rezolucije**.

Princip rezolucije je logički zasnovan i kompletan sa jednim ograničenjem:

Argumenti principa rezolucije su pojednostavljena verzija logike predikata, tzv. **klauzule**. Simboli, izrazi i atomi su isti kao u logici predikata, ali se koriste literali i klauzule umesto logičkih i kvantifikatorskih formula.

Klauzalni oblik (clausal form): pojednostavljena verzija logike predikata (ekvivalent)

Literal je atomska formula ili njena negacija.

Klauzula je skup literala koji predstavlja njihovu disjunkciju.

Princip **rezolucije** $\alpha \lor \beta, \neg \beta \lor \gamma \surd \alpha \lor \gamma$

ili

$P \lor Q$

$\neg P \lor R$

$Q \lor R$

Primena rezolucije:

Skupu aksioma i premisa se doda negacija teoreme, pa ako se dobije kontradikcija, teorema je dokazana. Takodje omogućava odgovaranje na pitanja.

Prevodjenje u klauzalni oblik:

1. Eliminisanje implikacije

$\alpha \Rightarrow \beta$ se zamenjuje sa $\neg \alpha \vee B$

2. Sužavanje oblasti delovanja negacije

$\neg \neg\alpha \equiv \alpha$

$\neg(\alpha \wedge \beta) \equiv \neg\alpha \vee \neg\beta$ $\qquad$ $\neg\forall x\ \alpha \equiv \exists x \neg\ \alpha$

$\neg(\alpha \vee \beta) \equiv \neg\alpha \wedge \neg\beta$ $\qquad$ $\neg\exists x\ \alpha \equiv \forall x \neg\ \alpha$

3. Standardizacija promenljivih

Promenljive se preimenuju tako da svaki kvatifikator ima jedinstvenu promenljivu.

$(\forall x\ P(x,x)) \wedge (\exists x\ Q(x))$ postaje $(\forall x\ P(x,x)) \wedge (\exists y\ Q(y))$

4. Eliminisanje egzistencijalnih kvantifikatora (Skolemizacija)

1) Ako $\exists$ nije u oblasti delovanja $\forall$, onda se uvodi Skolem konstanta:
$\exists x\, P(x)$ se zamenjuje sa $P(A)$, gde je A nova konstanta

2) Ako $\exists$ jeste u oblasti delovanja $\forall$, onda se uvodi Skolem funkcija:
$\forall y\, \exists x\, P(x)$ se zamenjuje sa $\forall y\, P(f(y))$, gde je f nova funkcija

5. Izbacivanje univerzalnih kvantifikatora

6. Prevodjenje formule u konjuktivnu normalnu formu

$\alpha \vee (\beta \wedge \gamma)$ se zamenjuje sa $(\alpha \vee \beta) \wedge (\alpha \vee \gamma)$

7. Eliminsanje konjukcije

$\alpha \wedge (\beta \vee \gamma)$ se zamenjuje sa $\{\alpha\}$ i $\{\beta, \gamma\}$ (skup klauzula)

8. Preimenovanje promenljivih

Nijedna promenljiva se ne pojavljuje u više od jedne klauzule.

Primer:

Prevesti u klauzulni oblik formulu:

$$\forall x\ (\forall y\ P(x, y)\ ) \Rightarrow \neg\ (\forall y\ Q(x, y) \Rightarrow R(x, y)\ )$$

// $\forall x$ na početku formule deluje na celu formulu

Rešenje:

1) $\forall x\ \neg(\forall y\ \ P(x, y)\ )\ \vee\ \neg(\forall y\ \neg Q(x, y) \vee R(x, y)\ )$

2) $\forall x\ (\exists y\ \neg P(x,y)\ ) \vee (\exists y\ \ Q(x, y) \wedge \neg R(x, y)\ )$

3) $\forall x\ (\exists y\ \neg P(x, y)\ ) \vee (\exists z\ \ Q(x, z) \wedge \neg R\ (x, z)\ )$

4) $\forall x\ (\neg P(x, f1(x)\ )\ ) \vee (\ Q(x, f2(x)\ ) \wedge \neg R(x, f2(x)\ )\ )$

5) $(\neg P(x, f1(x)\ )\ ) \vee (\ Q(x, f2(x)\ ) \wedge \neg R(x, f2(x)\ )\ )$

6) $(\neg P(x, f1(x)\ ) \vee Q(x, f2(x)\ )\ ) \wedge (\neg P(x, f1(x)\ ) \vee \neg R(x, f2(x)\ )\ )$

7) $\{\ \neg P(x, f1(x)\ ), Q(x, f2(x)\ )\ \}$

$\{\ \neg P(x, f1(x)\ ), \neg R(x, f2(x)\ )\ \}$

8) $\{\neg P(x1, f1(x1)\ ), Q(x1, f2(x1)\ )\ \}$

$\{\ \neg P(x2, f1(x2)\ ), \neg R(x2, f2(x2)\ )\ \}$

Primer: Prevesti u klauzulni oblik formulu:

$\forall x\ P(x) \Rightarrow ((\forall y\ P(y) \Rightarrow P(f(x, y))) \wedge \neg(\forall y\ Q(x, y) \Rightarrow P(y)))$

Deo rešenja:

// a + bc = (a+b) (a+c)

$\{P(x1), \neg P(y), P(f(x1,y))\}$
$\{\neg P(x2), Q(x2, g(x2))\}$
$\{\neg P(x3), \neg P(g(x3))\}$

Izvodjenje rezolucijom

Formula α se izvodi iz zadatog skupa klauzula tako sto se formira niz klauzula takav da je:

a) α element niza

b) svaki element je ili iz baze, ili je dobijen primenom principa rezolucije na klauzule ranije u nizu

Dokazivanje pomocu rezolucije (pobijanje, resolution refutation)

Skupu klauzula se doda negacija teoreme (prevedena u klauzalni oblik), i primenjuje se rezolucija. Ako se dobije prazna klauzula, onda teorema logicki sleduje iz polaznog skupa klauzula.

Moguce primene: (1) true-or-false pitanja, ili (2) fill-in-the-blank pitanja

Primer 1:

1. { P(x), Q(x, y) } …Δ
2. { ¬P(A), R(B, Z) }…Δ
3. { Q(A, y), R(B, Z) } …{ x / A } x se menja sa A, na osnovu 1. i 2.

Primer 2.

1. { P(x, y), Q(x), R(x) }…Δ
2. { ¬P(A, Z), ¬Q(B) }…Δ
3. { P(B, y), R(B), ¬P(A, Z) } … { x / B } iz 1. i 2.
4. { Q(A), R(A), ¬Q(B) } … { x / A, y / Z } iz 1. i 2.

Primer 3.

1. { P }…Δ
2. { ¬P }…Δ
3. { } ← Prazna klauzula

Prazna klauzula pokazuje da u polaznom skupu postoji kontradiktornost. Proces pobijanja je proces dokazivanja logičkog sledovanja klauzule iz baze. Postupak pobijanja: negacija ciljne klauzule doda se bazi i nakon toga se primenjuje rezolucija. Ako je rezultat primene rezolucije prazna klauzula formula važi.

Neke moguće primene rezolucije

1) Odgovaranje na pitanja sa DA ili NE

Koriste se dva predikata: O(x, y) – otac, R(x, y) – roditelj.

Treba pokazati da je Dragan Milanov roditelj.

O(Dragan, Milan)
O(Petar, Jasna)
O(x, y) => R(x, y)
Dokazati: R(Dragan, Milan)
(negacija teoreme: ¬R(Dragan, Milan) u klauzulni oblik)

1. { O(Dragan, Milan) } …Δ
2. { O(Petar, Jasna) } …Δ
3. { ¬O(x, y), R(x, y) } …Δ ⇔ O(x, y) => R(x, y)
4. { ¬R(Dragan, Milan) }…Γ
5. { R(Dragan, Milan) } … iz 1. i 3. { x / Dragan, y / Milan }
6. { R(Petar, Jasna) } … iz 2. i 3. { x / Petar, y / Jasna }
7. { ¬O(Dragan, Milan) } iz 3. i 4. { x / Dragan, y / Milan }
8. { } … iz 4. i 5. ←Ovde je dokaz gotov => DA
9. { } … iz 1. i 7. ←Ne mora i ovo

2) Odgovaranje na pitanja - “Ko je Milanov otac ? “ , tj određivanje veze za promenljivu

1. { O(Dragan, Milan) } …Δ
2. { O(Petar, Jasna) } …Δ
3. { ¬O(x, y), R(x, y) } …Δ
4. { ¬R(z, Milan), **Odg(z)** } … Γ // Ovde je dodatni literal ‘Odgovor’

Postupak je isti osim što se u dodatnu klauzulu doda literal odgovor. Postupak prestaje kada se dobije klauzula ‘Odg’ (odgovor).

5. { R(Dragan, Milan) } … iz 1. i 3. { x / Dragan, y / Milan }
6. { R(Petar, Jasna) } … iz 2. i 3. { x / Petar, y / Jasna }
7. { ¬O(w, Milan), Odg(w) } … iz 3. i 4. { x / w, z / w, y / Milan }
8. { Odg(Dragan) } … iz 4. i 5.
9. { Odg(Dragan) } … iz 1. i 7.

Primer: Ako je glava ja dobijam, ako je pismo ti gubiš. Koristeći rezoluciju pokazati da ja dobijam.

Rešenje:

Predikati: G – gubiš, D – dobijam, GL – glava, P – pismo, D(Ja) – teorema.

Rečenice logike predikata:

GL => D(Ja)

P => G(Ti)

¬GL => P

G(Ti) => D(Ja)

Klauzulni oblik i rezolucija:

1. {¬GL, D(Ja)}…Δ
2. {¬P, G(Ti)}…Δ
3. {GL, P}…Δ
4. {¬G(Ti), D(Ja)}…Δ
5. {¬D(Ja)}…Δ
6. {¬P, D(Ja)}… iz 2. i 4.
7. {P, D(Ja)}… iz 2. i 3.
8. {D(Ja)}… iz 6. i 7.
9. {}… iz 5. i 8. => Teorema važi!

Strategije rezolucije

a) **Čista rezolucija** (proba svako sa svakim, nema strategije)

Jednostavne strategije rezolucije

b) **Eliminacija čistih literala** - čist literal je onaj koji nema odgovarajući komplementarni literal u polaznoj bazi. Klauzule koje sadrže čiste literale se izbacuju. Ova strategija se primenjuje samo jednom i to na početku.

Primer.

{ ¬P, ¬Q, R }

{ ¬P, S })

{ ¬Q, S }) ← Izbacuju se ove dve klauzule zato što je S čist literal

{ P }

{ Q }

{ ¬R }

c)**Eliminisanje tautologija** – tautologija je klauzula koja sadrži par komplementarnih literala. Ova strategija se ne primenjuje ukoliko su literali komplementarni nakon primene unifikacije. Može da se primenjuje više puta u toku rada.

Primer.

{ P(F(A)), $\neg$P(F(A)) }

{ P(x), Q(y), $\neg$Q(y), R(z) }

- izbacuju se

d) **Eliminisanje podrazumevanih klauzula** – klauzula Φ podrazumeva klauzulu Ψ. Akko postoji zamena σ tako da je $\Phi\sigma \subseteq \Psi$ i u tom slučaju Ψ može da se izbaci. Može se primeniti više puta.

Primer.

{ P(x), Q(y) } ← Podrazumeva donju klauzulu pa se donja može izbaciti

{ P(A), Q(v), R(w) }

Opšte strategije rezolucije

e) **jedinicna rezolucija** (unit resolution)

Preferira klauzule koje sadrze samo jedan literal (jedinična kluzula).

Teži se da bar jedan roditelj bude jedinična klauzula.

f) **ulazna rezolucija** (input resolution)

Svaka primena rezolucije ukljucuje jednu od ulaznih klauzula (od zadatih iz baze znanja, ili iz upita) - barem jedan roditelj je član polazne (ulazne) baze. Ova strategija rezolucije je ekvivalentna jedinicnoj rezoluciji, i kompletna je za Hornove klauzule.

g) **Linearna rezolucija** (ancestry-filtered)

Modifikacija ulazne rezolucije – barem jedna klauzula je ili iz baze znanja ili je prethodnik druge klauzule.

h) set-of-support resolution

Barem jedna klauzula potice iz negacije teoreme, ili je iz nje izvedena.

i) Uređena rezolucija

Svaka klauzula se tretira kao linearno uređeni skup. Rezolucija se dozvoljava samo na prvom literalu. U rezultatu najpre se pišu literali iz pozitivnog roditelja, a iza njih su literali iz negativnog roditelja. Kompletna je samo za Hornove klauzule.

j) Usmerena rezolucija

Usmerena klauzula je Hornova klauzula u kojoj se pozitivni literal javlja kao prvi ili kao poslednji literal.
Ako se javlja kao prvi literal, onda je to backward clause.
Ako je poslednji, forward clause.
U usmerenoj rezoluciji, razmatra se samo prvi literal (ili poslednji).

k) subsumption elimination

Eliminisu se sve klauzule koje su specificnije od postojecih klauzula u bazi znanja.

l) Eliminacija tautologija

Eliminisu se sve klauzule koje predstavljaju tautologiju.

Primer: Čista rezolucija, 21 korak za dobijanje prazne klauzule:

1) $\{P, Q\}$… Δ
2) $\{\neg P, R\}$… Δ
3) $\{\neg Q, R\}$… Δ
4) $\{\neg R\}$… Γ
5) $\{Q, R\}$… iz 1) i 2)
6) $\{P, R\}$… iz 1) i 3)
7) $\{\neg P\}$… iz 2) i 4)
8) $\{\neg Q\}$… iz 3) i 4)
9) $\{P\}$… iz 1) i 8)
10) $\{R\}$… iz 2) i 6)
...
21) $\{\}$

Primer: Jedinična rezolucija

1) $\{P, Q\} \ldots \Delta$
2) $\{\neg P, R\} \ldots \Delta$
3) $\{\neg Q, R\} \ldots \Delta$
4) $\{\neg R\} \ldots \Gamma$
5) $\{\neg P\} \ldots$ 2) i 4)
6) $\{\neg Q\} \ldots$ 3) i 4)
7) $\{Q\} \ldots$ 1) i 5)
8) $\{P\} \ldots$ 1) i 6)
9) $\{R\} \ldots$ 2) i 8)
10) $\{R\} \ldots$ 3) i 7)
11) $\{\} \ldots$ 5) i 8)

Primer: Ulazna rezolucija

1) {P, Q}… Δ
2) {¬P, R}… Δ
3) {¬Q, R}… Δ
4) {¬R}… Γ
5) {Q, R } … 1) i 2)
6) {P, R } … 1) i 3)
7) {¬P } … 2) i 4)
8) {¬Q } … 3) i 4)

...

19) { }

Primer: Linearna rezolucija

1) $\{P, Q\} \ldots \Delta$
2) $\{\neg P, R\} \ldots \Delta$
3) $\{\neg Q, R\} \ldots \Delta$
4) $\{\neg R\} \ldots \Gamma$

$\{P, Q\}$ $\{\neg P, Q\}$ $\{P, \neg Q\}$ $\{\neg P, \neg Q\}$

$\{Q\}$

$\{P\}$ $\{Q\}$

$\{\neg Q\}$

$\{\ \}$

Primer: Rezolucija na osnovu skupa podrške

1) {P, Q}… Δ
2) {¬P, R}… Δ
3) {¬Q, R}… Δ
4) {¬R}… Γ
5) {¬P } … 2) i 4) // zbog 4) se radi 7.
6) {¬Q } … 3) i 4)
7) {Q } … 1) i 5) // 5) je potomak negacije teoreme
8) {P } … 1) i 6) // 6) je potomak negacije teoreme
9) {R }… 3) i 7)
10) { } … 6) i 7)

Primer: Uređena rezolucija –

1) $\{P, Q\} \ldots \Delta$
2) $\{\neg P, R\} \ldots \Delta$
3) $\{\neg Q, R\} \ldots \Delta$
4) $\{\neg R\} \ldots \Gamma$
5) {Q, R } … 1) i 2)
6) { R } … 3) i 5)
7) { } … 4) i 6)