Prof.dr.sc. Bojana Dalbelo Bašić

Fakultet elektrotehnike i računarstva
Zavod za elektroniku, mikroelektroniku, računalne i inteligentne sustave

www.zemris.fer.hr/~bojana
bojana.dalbelo@fer.hr

UMJETNA INTELIGENCIJA

Zaključivanje uporabom predikatne logike (2)

REZOLUCIJSKO ZAKLJUČIVANJE

- Nastavljamo s teorijom dokaza

PREDUVJETI ZA REZOLUCIJSKO ZAKLJUČIVANJE

Pretvaranje formule u klauzalni oblik

- Rezolucijsko pravilo u predikatnoj logici zahtijeva pretvaranje formule u **klauzalni oblik**
- Implicitno se podrazumijeva:
  - sve varijable u klauzuli su **univerzalno kvantificirane**
  - između klauzula je konjunkcija
- Sve klauzule u predikatnoj logici su **standardizirane** – ne postoje dvije klauzule koje sadrže iste **varijable**
- Pretvorba u klauzalni oblik kroz **10 slijednih koraka**

PRETVARANJE FORMULE U KLAUZALNI OBLIK

1. **Uklanjanje $\leftrightarrow$**
   - $(F \leftrightarrow G) \equiv (\sim F \vee G) \wedge (\sim G \vee F)$
2. **Uklanjanje $\rightarrow$**
   - $(F \rightarrow G) \equiv (\sim F \vee G)$
3. **Smanjivanje dosega operatora** negacije tako da se odnosi samo na jedan atom
   - $\sim(F \vee G) \equiv (\sim F \wedge \sim G)$
   - $\sim(F \wedge G) \equiv (\sim F \vee \sim G)$
   - $\boldsymbol{\sim\forall x\, F(x) \equiv \exists x\, (\sim F(x))}$
   - $\boldsymbol{\sim\exists x\, F(x) \equiv \forall x\, (\sim F(x))}$

Kada se u nekom od prethodna tri koraka pojavi dvostruka negacija, eliminiraj je primjenom **involutivnosti** $\boldsymbol{(\sim(\sim F)) \equiv F}$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

4. **Preimenuj varijable** tako da svaki kvantifikator vezuje jedinstvenu varijablu.

- $(\forall x\, F(x) \lor \forall x\, G(x)) \equiv (\forall x\, F(x) \lor \forall y\, G(y))$
- $(\forall x\, F(x) \lor \exists x\, G(x)) \equiv (\forall x\, F(x) \lor \exists y\, G(y))$
- $(\exists x\, F(x) \lor \forall x\, G(x)) \equiv (\exists x\, F(x) \lor \forall y\, G(y))$
- $(\exists x\, F(x) \lor \exists x\, G(x)) \equiv (\exists x\, F(x) \lor \exists y\, G(y))$
- $(\forall x\, F(x) \land \forall x\, G(x)) \equiv (\forall x\, F(x) \land \forall y\, G(y))$
- $(\forall x\, F(x) \land \exists x\, G(x)) \equiv (\forall x\, F(x) \land \exists y\, G(y))$
- $(\exists x\, F(x) \land \forall x\, G(x)) \equiv (\exists x\, F(x) \land \forall y\, G(y))$
- $(\exists x\, F(x) \land \exists x\, G(x)) \equiv (\exists x\, F(x) \land \exists y\, G(y))$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

5. **Skolemizacija**
   - Zamijeni sve egzistencijalno kvantificirane varijable *Skolem-izrazima*

Primjer:

- ∃x SESTRA(x, Ivan);

Skolemizacija daje:

- SESTRA (Ana, Ivan)

Skolem-konstanta

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- U složenijim izrazima u kojima vrijednost zamjene zavisi od ostalih varijabli u formuli, egzistencijalno kvantificirane varijable zamjenjuju se tzv. SKOLEM-FUNKCIJOM

Primjer:

- U formuli $\forall x\ \exists y$ MAJKA(y, x);
  - vrijednost od *y* zavisi o *x*.

- Skolemizacija daje
  - $\forall x$ MAJKA ($f(x)$, x),

    gdje je $f(x)$ Skolem funkcija

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- Argumenti Skolem-funkcije su one univerzalno kvantificirane varijable čiji doseg uključuje doseg egzistencijalno kvantificirane varijable koja se zamjenjuje

Primjer:

- $\exists u \, \forall v \, \forall w \, \exists x \, \forall y \, \exists z \; F(u, v, w, x, y, z);$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- Argumenti Skolem-funkcije su one univerzalno kvantificirane varijable čiji doseg uključuje doseg egzistencijalno kvantificirane varijable koja se zamjenjuje

Primjer:

- $\exists \boldsymbol{u}\ \forall v\ \forall w\ \exists \boldsymbol{x}\ \forall y\ \exists \boldsymbol{z}\ F(u, v, w, x, y, z)$;

  Eliminiraju se $\exists u$, $\exists x$, i $\exists z$:

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- Argumenti Skolem-funkcije su one univerzalno kvantificirane varijable čiji doseg uključuje doseg egzistencijalno kvantificirane varijable koja se zamjenjuje

Primjer:

- $\exists \boldsymbol{u}\ \forall v\ \forall w\ \exists \boldsymbol{x}\ \forall y\ \exists \boldsymbol{z}\ F(u, v, w, x, y, z);$

  Eliminiraju se $\exists u$, $\exists x$, i $\exists z$:

  $\forall v\ \forall w\ \forall y\ F(\boldsymbol{a}, v, w, \boldsymbol{f(v, w)}, y, \boldsymbol{g(v, w, y)});$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- Argumenti Skolem-funkcije su one univerzalno kvantificirane varijable čiji doseg uključuje doseg egzistencijalno kvantificirane varijable koja se zamjenjuje

Primjer:

- $\exists \boldsymbol{u}\ \forall v\ \forall w\ \exists \boldsymbol{x}\ \forall y\ \exists \boldsymbol{z}\ F(u, v, w, \boldsymbol{x}, y, \boldsymbol{z})$;
- Eliminiraju se $\exists u$, $\exists x$, i $\exists z$:

  $\forall v\ \forall w\ \forall y\ F(\boldsymbol{a}, v, w, \boldsymbol{f(v, w)}, y, \boldsymbol{g(v, w, y)})$;

  i zamjenjuju redom SKOLEM-IZRAZIMA:

  a, *f*(v, w), *g*(v, w, y), gdje je *a* Skolem-konstanta dok su *f*, *g* Skolem-funkcije. .

- Niti jedan od simbola a, f, g ne smije se pojavljivati u izvornoj formuli

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- Skolemizacija kao postupak ne daje nužno definiciju Skolem-funkcije nego se radi o metodi pridjeljivanja imena funkcijama koje moraju postojati

Primjer:

- $\forall x\ \exists y\ GT(y, x)$
- Skolemizacija daje $\forall x\ GT(f(x), x)$,
- *f* je Skolem-funkcija koja može biti:
  - $f(x) = x + 1$ ili
  - $f(x) = x + 5$ itd.

PRETVARANJE FORMULE U KLAUZALNI OBLIK

- **OPRAVDANJE ZA SKOLEMIZACIJU?**
  - Zašto egzistencijalno kvantificiranu varijablu smijemo zamijeniti konstantom?
  - $\exists x$ SESTRA(x, Ivan) $\equiv$ SESTRA(Ana, Ivan) **???**
- Odgovor:
  - gornja ekvivalencija općenito ne vrijedi, ali...
  - **... skolemizacija ne utječe na svojstvo <u>nezadovoljivosti</u> formule!**
  - Ako $\exists$**x** SESTRA(**x**, Ivan) $\equiv \perp$

    onda SESTRA(**Ana**, Ivan) $\equiv \perp$
  - U kontekstu rezolucije opovrgavanjem to je dovoljno
    - Ako su premise + negirani cilj proturječni, bit će takvi i nakon skolemizacije

PRETVARANJE FORMULE U KLAUZALNI OBLIK

6. Premjesti sve kvantifikatore (preostali su samo univerzalni) na lijevu stranu formule tako da se na lijevoj strani nalazi niz kvantifikatora koji se nazivaju **prefiks**. Desna strana formule koja se naziva **matrica**, oslobođena je svih kvantifikatora.

   Međusobni uređaj kvantifikatora ostaje nepromijenjen! Formula u takvom obliku se naziva **PRENEKS NORMALNI OBLIK**

- $(\forall x\, F(x) \vee \forall y\, G(y)) \equiv \forall x\, \forall y\, (F(x) \vee G(y))$
- $(\forall x\, F(x) \wedge \forall y\, G(y)) \equiv \forall x\, \forall y\, (F(x) \wedge G(y))$
- $(\forall x\, F(x) \vee H\{x\}) \equiv \forall x\, (F(x) \vee H\{x\})$
- $(\forall x\, F(x) \wedge H\{x\}) \equiv \forall x\, (F(x) \wedge H\{x\})$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

7. **Eliminiraj prefiks**, ostavi samo matricu. Podrazumijeva se da su sve varijable u formuli univerzalno kvantificirane. (Nema slobodnih varijabli u formuli)

8. Pretvori matricu u **konjunkciju disjunkcija** (konjunktivnu normalnu formu) koristeći distributivnost
   - $(F \vee (G \wedge H)) \equiv ((F \vee G) \wedge (F \vee H))$
   - $((G \wedge H) \vee F) \equiv ((G \vee F) \wedge (H \vee F))$

9. Napiši konjunktivnu normalnu formu kao **skup klauzula** brišući operatore konjunkcija. Implicitno se podrazumijava konjunkcija između klauzula

PRETVARANJE FORMULE U KLAUZALNI OBLIK

10. **Standardiziraj** klauzule preimenovanjem varijabli tako da nema dvije klauzule koje sadrže iste varijable
    - Sve varijable u klauzulama implicitno su kvantificirane (korak 7) i postoji implicitna konjunkcija između klauzula (korak 9) pa je preimenovanje varijabli valjano i dopušteno jer se temelji na slijedećoj ekvivalenciji:

$$\forall x\,(F(x) \land G(x)) \equiv \forall x\,\forall y\,(F(x) \land G(y))$$

- **<u>Pazi</u>!** Svrha preimenovanja nije ukloniti višestruko pojavljivanje iste varijable **unutar iste klauzule**!
    - Npr. općenito **ne vrijedi**:

$$\forall x\,P(x,x) \equiv \forall x\,\forall y\,P(x,y)$$

$$\forall x\,(P(x) \lor Q(x)) \equiv \forall x\,\forall y\,(P(x) \lor Q(y))$$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

Primjer

- Pretvorba formule u klauzalni oblik

  $\forall y\ \forall z\ (\exists u(P(y, u) \vee P(z,u)) \rightarrow \exists u Q(y, z, u))$

1. **Uklanjanje $\leftrightarrow$**

   OK

2. **Uklanjanje $\rightarrow$**

   $\forall y\ \forall z\ (\sim(\exists u\ (P(y, u) \vee P(z,u))) \vee \exists u Q(y, z, u))$

3. **Smanjivanje dosega operatora ~**

   $\forall y\ \forall z\ (\forall u(\sim P(y, u) \wedge \sim P(z,u)) \vee \exists u Q(y, z, u))$

4. **Preimenovanje varijabli**

   $\forall y\ \forall z\ (\forall u(\sim P(y, u) \wedge \sim P(z,u)) \vee \exists v Q(y, z, v))$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

Primjer

- Pretvorba formule u klauzalni oblik

  $\forall y\ \forall z\ (\exists u(P(y, u) \vee P(z,u)) \rightarrow \exists u Q(y, z, u))$

1. **Uklanjanje $\leftrightarrow$**

   OK

2. **Uklanjanje $\rightarrow$**

   $\forall y\ \forall z\ (\sim(\exists u\ (P(y, u) \vee P(z,u))) \vee \exists u Q(y, z, u))$

3. **Smanjivanje dosega operatora ~**

   $\forall y\ \forall z\ (\forall u(\sim P(y, u) \wedge \sim P(z,u)) \vee \exists u Q(y, z, u))$

4. **Preimenovanje varijabli**

   $\boldsymbol{\forall y\ \forall z\ (}\forall u(\sim P(y, u) \wedge \sim P(z,u)) \vee \boldsymbol{\exists v} Q(y, z, \boldsymbol{v})\boldsymbol{)})$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

5. **Skolemizacija**

$$\forall y\ \forall z\ (\forall u(\sim P(y, u) \land \sim P(z,u)) \lor Q(y, z, f(y,z)))$$

6. **Preneks normalni oblik**

$$\forall y\ \forall z\ \forall u((\sim P(y, u) \land \sim P(z,u)) \lor Q(y, z, f(y,z)))$$

7. **Eliminiraj prefiks**

$$(\sim P(y, u) \land \sim P(z,u)) \lor Q(y, z, f(y,z))$$

8. **Konjunktivni normalni oblik**

$$(\sim P(y, u) \lor Q(y, z, f(y, z))) \land$$
$$(\sim P(z, u) \lor Q(y, z, f(y, z)))$$

PRETVARANJE FORMULE U KLAUZALNI OBLIK

9. **Skup klauzula**

   $\sim P(y, u) \vee Q(y, z, f(y, z))$

   $\sim P(z, u) \vee Q(y, z, f(y, z))$

10. **Standardizacija (nakon** u->v, y->w, z->x),

   $\sim P(y, u) \vee Q(y, z, f(y, z))$

   $\sim P(x, v) \vee Q(w, x, f(w, x))$

ZAMJENA (SUPSTITUCIJA)

- Univerzalno kvantificirane varijable zamjenjivali smo konstantom.

Primjer

  - [1] $\forall x(MUŽ(x) \rightarrow VOLI(x, ŽENA(x)))$
  - [2] MUŽ(Marko)
  - [3] (MUŽ(Marko) $\rightarrow$ VOLI(Marko, ŽENA(Marko)).
- Općenito varijable mogu biti **zamijenjene (supstituirane)** čitavim **izrazima**
- **Supstitucija** je neophodna za primjenu rezolucijskog pravila

ZAMJENA (SUPSTITUCIJA)

Skup uređenih parova čini **zamjenu** (supstituciju)

α **= {t_1/y_1, t_2/y_2, … , t_n/y_n,},**

gdje su t_i, i = 1, 2, … n izrazi,

y_i, i = 1, 2, … n su <u>različite</u> varijable, i vrijedi $t_i \neq y_i$

za bilo koji i = 1, 2, … n.

Primjer:

- izraz GT(x, y),
  - supstitucija α = {prosinac/x, travanj/y},
  - rezultat Kα = GT(prosinac, travanj).

ZAMJENA (SUPSTITUCIJA)

- Kad se zamjena α primjeni na neki izraz K, tada se svako pojavljivanje y_i u K istodobno zamijeni sa t_i.

- Dobiveni izraz označavamo sa Kα i kažemo da je **Kα slučaj** ili primjer izraza K (*engl. instance*).

- **Prazna zamjena ε** je takva da je Kε = K

Primjer:

izraz K = P(w, f(x), z),

- supstitucija α = {a/x, g(u)/z},
- rezultat Kα = P(w, f(a), g(u))

KOMPOZICIJA ZAMJENA

- Neka su α i β dvije zamjene. **Kompozicija zamjena** je definirana kao nova zamjena $\alpha \quad \beta$ tako da je

  K($\alpha \quad \beta$) = (Kα)β, za neki izraz K

- Uporaba zamjene $\alpha \quad \beta$ na izrazu K isto kao prvo α na K, a onda se na tako dobiveni izraz primjeni β.

- Kompozicija zamjena $\alpha \quad \beta$ može biti izvedena na temelju sljedećeg postupka u dva koraka

KOMPOZICIJA ZAMJENA

- Neka su dane α i β zamjene

  $\alpha = \{t_1/y_1, t_2/y_2, \dots , t_n/y_n\}$,

  $\beta = \{s_1/x_1, s_2/x_2, \dots , s_m/x_m\}$

<u>1. korak</u>

- Iz α i β konstruiraj skupove $\lambda 1$, $\lambda 2$, $\lambda 3$:

  $\lambda 1 = \{ t_1\beta/y_1, t_2\beta/y_2, \dots , t_n\beta/y_n, s_1/x_1, s_2/x_2, \dots , s_m/x_m\}$

  $\lambda 2 = \{ t_i\beta/y_i \mid t_i\beta/y_i \in \lambda 1$ i $t_i\beta = y_i$ za $1 \le i \le n\}$,

  $\lambda 3 = \{ s_i/x_i \mid s_i/x_i \in \lambda 1$ i $x_i \in \{ y_1, y_2, \dots , y_n\}\}$;

<u>2. korak</u>

- Izvedi kompoziciju zamjena $\alpha \quad \beta$ uporabom formule

$$\alpha \quad \beta = \lambda 1 - \lambda 2 - \lambda 3$$

KOMPOZICIJA ZAMJENA - PRIMJER

Primjer

- Neka su zadane zamjene

  α = {z/u, h(u)/w}

  β = {a/u, z/w, u/z}

- Izvedi kompoziciju ($\alpha \quad \beta$) na temelju postupka u dva koraka, a zatim pokaži da vrijedi K($\alpha \quad \beta$) = (Kα)β za izraz **K = P(u, w, f(z))**

<u>1. korak</u> – konstrukcija skupova λ1, λ2 i λ3.

- λ1 = {zβ/u, h(u)β/w, a/u, z/w, u/z}

  = {<u>u/u,</u> h(a)/w, <u>a/u, z/w</u>, u/z}

- λ2 = {u/u}
- λ3 = {a/u, z/w}

<u>2. korak</u> – izvođenje kompozicije zamjena $\alpha \quad \beta$ uporabom formule

- $\alpha \quad \beta$ = λ1 - λ2 - λ3 = {h(a)/w, u/z}

KOMPOZICIJA ZAMJENA - PRIMJER

- Još treba pokazati da se isti rezultat dobije primjenom definicije zamjena tj. da je $K(\alpha \quad \beta) = (K\alpha)\beta$ za slučaj $K = P(u, w, f(z))$

  $\alpha = \{z/u, h(u)/w\}$

  $\beta = \{a/u, z/w, u/z\}$
- (i) prema definiciji:
  - $K\alpha = P(z, h(u), f(z))$
  - $(K\alpha)\beta = \mathbf{P(u, h(a), f(u))}$
- (ii) primjenom izvedene formule za $\alpha \quad \beta$:

  $\alpha \quad \beta = \{h(a)/w, u/z\}$

  $K(\alpha \quad \beta) = \mathbf{P(u, h(a), f(u))}$
- Dakle vrijedi $K(\alpha \quad \beta) = (K\alpha)\beta$

SVOJSTVA KOMPOZICIJE ZAMJENA

Asocijativnost

- $(\alpha \quad \beta) \quad \gamma = \alpha \quad (\beta \quad \gamma)$

ε je neutralni element s lijeva

- $\varepsilon \quad \alpha = \alpha$

ε je neutralni element s desna

- $\alpha \quad \varepsilon = \alpha$

Općenito ne vrijedi komutativnost

- $\alpha \quad \beta \neq \beta \quad \alpha$

UNIFIKACIJA

- Općeniti izrazi K1 i K2 mogu se svesti na isti oblik akko postoji supstitucija γ takva da vrijedi

$$\boxed{K_1 \gamma = K_2 \gamma}$$

- Supstitucija γ se naziva **unifikator** (*engl. unifier*)
- Izraz $K_i \gamma$, i =1, 2 naziva se **zajednički slučaj** ili primjer *(engl. common instance*)
- Kaže se da su opći izrazi K1 i K2 unificirani pomoću γ
- Izrazi mogu imati više unifikatora

UNIFIKACIJA - PRIMJER

- Atomi **P(x) i P(y),** gdje su x i y varijable, imaju unifikatore:
  - γ_1 **= {b/x, b/y}** koji daje zajednički slučaj P(b) i
  - γ_2 **= {z/x, z/y}** koji daje zajednički slučaj P(z).
- Ako je ***b*** konstanta, a ***z*** varijabla možemo reći da je P(z) <u>općenitiji zajednički slučaj </u>nego li je to P(b)
- Naime, P(b) je slučaj od P(z) uz supstituciju {b/z}, ali NE VRIJEDI OBRATNO. (**Zašto**?)

NAJOPĆENITIJI UNIFIKATOR

- Za gornji primjer kažemo da je unifikator $\gamma 2$ **općenitiji** od unfikatora $\gamma 1$ za atome P(x) i P(y). Naime, tada postoji zamjena $\boldsymbol{\gamma 3}$ takva da je

$$(P\ \boldsymbol{\gamma 2})\ \boldsymbol{\gamma 3} = P\ \boldsymbol{\gamma 1}$$

- P(x) i P(y) mogu imati više unifikatora – zanima nas najopćenitiji unifikator
- Unifikator δ se naziva **najopćenitiji unifikator** akko za svaki unifikator γ od K_1 i K_2, zajednički primjer $K_i\delta$ je općenitiji (ili jednak) od zajedničkog primjera $K_i\gamma$
- Izraz $K_i\delta$ se naziva **najopćenitiji zajednički slučaj**

  \- $K_i\gamma$ je primjer od $K_i\delta$

NAJOPĆENITIJI UNIFIKATOR

δ je **najopćenitiji unifikator** akko za svaki unifikator γ od K_i, i=1,2 postoji zamjena θ takva da je

$$\mathbf{K_i}\,\gamma = (\mathbf{K_i}\,\delta)\,\theta = K_i\,(\delta \quad \theta)$$

$$\text{tj. } \gamma = \delta \quad \theta.$$

- MGU se može tumačiti kao da ima za posljedicu najmanje moguće promjene da bi se unificirali izrazi
- Pretpostavnimo da tijekom postupka nalaženja zajedničkog primjera dva izraza postoji izbor zamijeniti varijablu konstantom ili zamijeniti varijablu drugom varijablom. Procedura nalaženja MGU će uvijek izabrati ovo drugo
- Dva izraza mogu imati više najopćenitijih unifikatora – u tom su slučaju svi najopćenitiji zajednički izrazi jednako općeniti

ALGORITAM ZA UNIFIKACIJU IZRAZA

- U literaturi postoji više poznatih algoritama za nalaženje najopćenitijeg unifikatora (dojavljuju pogrešku ako se izrazi ne mogu unificirati)
- Razmotrit ćemo rekurzivni algoritam MGUNIFIER (Luger, Stubblefield, 1993; Shinghal, 1992)
- Algoritam koristi pojmove:
  - glava općenitog izraza
  - rep općenitog izraza
- Općeniti izraz možemo napisati kao **listu** gdje su elementi liste odvojeni prazninama, a svaki element liste je opet moguća lista nazvana podizraz

ALGORITAM ZA UNIFIKACIJU IZRAZA

Primjer

- **P(x, y, f(b))** može biti napisano kao lista **(P x y (f b))** gdje su element liste P, x, y, (f b).
  Zadnji element liste je i sam lista, sastavljen od elemenata f i b
- **Glava liste** (P x y (f b)) je P, dok je **rep liste** (x y (f b))
- Sintaksa lista je sintaksa programskog jezika LISP

Sintaksa FOPL	***Sintaksa liste***
P(a, b)	(P a b)
P(f(a), g(x,y))	(P (f a) (g x y))
EQUAL(Eva, MAJKA(Iva))	(EQUAL Eva (MAJKA Iva))
P(x) $\wedge$ Q(y)	((P x) $\wedge$ (Q y))

REKURZIVNI ALGORITAM ZA NALAŽENJE NAJOPĆENITIJEG ZAJEDNIČKOG UNIFIKATORA

procedura MGUNIFIER (K_1, K_2)

ako ili K_1 ili K_2 simbolizira <u>konstantu, varijablu, funkciju, predikat</u> ili praznu listu **tada**:

ako K_1 i K_2 identični, **tada** vrati praznu supstituciju {};

ako K_1 ili K_2 prazne liste, **tada** vrati pogrešku;

ako K_1 predstavlja varijablu **tada**:

ako se K_1 pojavljuje u K_2 **tada** vrati *pogreška*;

inače vrati $\{K_2 / K_1\}$;

ako K_2 predstavlja varijablu **tada**:

ako se K_2 pojavljuje u K_1 **tada** vrati *pogreška*;

inače vrati $\{K_1 / K_2\}$;

ako niti K_1 niti K_2 ne predstavlja varijablu **tada** vrati *pogreška*;

inače

α := MGUNIFIER(glava od K_1 , glava od K_2);

ako α = *pogreška* **tada** vrati *pogreška*;

REKURZIVNI ALGORITAM ZA NALAŽENJE NAJOPĆENITIJEG ZAJEDNIČKOG UNIFIKATORA

… *(nastavak)*

K_3 := rezultat primjene zamjene α na rep od K_1;

K_4 := rezultat primjene zamjene α na rep od K_2;

β := MGUNIFIER(K_3, K_4);

ako β = *pogreška* **tada** vrati *pogreška*;

vrati kompoziciju zamjena α β ;

REKURZIVNI ALGORITAM ZA NALAŽENJE NAJOPĆENITIJEG ZAJEDNIČKOG UNIFIKATORA

- Procedura MGUNIFIER vraća pogrešku u slučajevima kada unifikacija nije moguća:
  - varijabla treba biti zamjenjena izrazom koji ju sadrži. Provjerava se pojavljuje li se varijabla u izrazu kojim se zamjenjuje. Takva zamjena vodila bi beskonačnoj petlji.
  - kada bi trebalo zamijeniti konstantu, funkciju ili predikat – što je u suprotnosti s definicijom supstitucije

PRIMJER

- Dani su literali

  P(x, x) i P(f(z), z).

- MGUNIFIER će prvo naći supstituciju {f(z)/x} koja daje

  P(f(z), f(z)) i P(f(z), z).

- <u>Kada ne bi bilo provjere uvjeta</u>, sljedeća supstitucija {f(z)/z}

  bi dala P(f(f(z)), f(f(z))) i P(f(f(z)), f(z))

  i tako u beskonačnost

PRIMJER

- Primjeri izraza koji se ne mogu unificirati

K1	K2	Uzrok pogrešci kod unifikacije
P(x)	P(f(x))	Vidi prethodni primjer (beskonačna petlja)
P(f(x))	P(x)	
P(x)	Q(x)	Predikat se ne može zamijeniti drugim predikatom.
P(f(x))	P(g(x))	Funkcija se ne može zamijeniti drugom funkcijom.

PRIMJER

- Nađi MGU izraza:

  K1 = P(g(u), z, f(z))   i   K2 = P(x,y,f(b))

{g(u)/x} unificira prve podizraze od K1 i K2 koji se ne slažu

K1{g(u)/x} = P(**g(u)**, z, f(z))

K2{g(u)/x} = P(**g(u)**, y, f(b))

{y/z} unificira sljedeće izraze koji se ne slažu

Kompozicija {g(u)/x}   {y/z} = {g(u)/x, y/z}

K1{g(u)/x, y/z} = P(**g(u), y**, f(y))

K2{g(u)/x, y/z} = P(**g(u), y** ,f(b))

{b/y} unificira sljedeće izraze koji se ne slažu

Kompozicija {g(u)/x, y/z}   { b/y} = {g(u)/x, b/z, b/y}

je **MGU**, koji označavamo s δ.

- Primjenom mgu **δ = {g(u)/x, b/z, b/y}** na K1 i K2 dobivamo najopćenitiji zajednički primjer

  P(g(u), b, f(b))

PRIMJER

- Ako za neka 2 izraza MGU nije jedinstven, tj. postoji više najopćenitijih unifikatora, odgovarajući najopćenitiji zajednički izrazi su **alfabetske varijante** – razlikuju se samo po *imenima* varijabli
- *Primjer*

Skup izraza	Najopćenitiji zajednički primjer
{P(x), P(a)}	P(a)
{P(f(x), y, g(y)), P(f(x), z, g(x)}	P(f(x), x, g(x))
{P(f(x, g(a,y)), g(a, y)), P(f(x,z), z)}	P(f(x, g(a, y)), g(a, y))

UNIFIKACIJA LITERALA

Definicija

- Dva literala mogu se **unificirati** akko
  1. oba označavaju negirane atome ili oba označavaju afirmativne atome
  2. ti atomi se mogu unificirati.

Primjer

P(x) i P(y) ili ~P(x) i ~P(y)

Definicija

- Dva literala mogu se **komplementarno unificirati** akko
  1. jedan od njih je negirani atom, a drugi je afirmativni atom
  2. ti se atomi mogu unificirati.

Primjer

P(x) i ~P(y) ili ~P(x) i P(y)

- Rezolucija u FOPL slična je rezoluciji u propozicijskoj logici
- Neka su L_k, $k = 1, \ldots, i$ i M_l, $l = 1, \ldots, j$ literali (neki negirani, a neki afirmativni)
- Neka su dane dvije klauzule

  (I1) $L_1 \vee L_2 \vee \ldots \vee L_i$

  (I2) $M_1 \vee M_2 \vee \ldots \vee M_j$
- Prije primjene rezolucje klauzule trebaju biti standardizirane (preimenovati varijable)
- Pretpostavimo da u (I1) i (I2) postoje literali koji se mogu komplemenarno unificirati. Pretpostavimo (bez gubitka općenitosti – vrijedi komutativnost) da su to literali L_1 i M_1
- Pretpostavimo da se unifikacija L_1 i M_1 može obaviti sa **MGU** δ.

- Sada možemo razriješiti *roditeljske klauzule* (I1) i (I2) po L_1 i M_1 i izvesti *resolventnu klauzulu*

(I3) $$\underbrace{L_2\delta \vee L_3\delta \vee \ldots\ldots \vee L_i\delta}_{\text{Ostatak (I1)}} \vee \underbrace{M_2\delta \vee M_3\delta \vee \ldots\ldots \vee M_j\delta}_{\text{Ostatak (I2)}}$$

- Literali u resolventi dobiju se primjenom $\vee$ na uniju literala u roditeljskim klauzulama izuzevši literale s kojima se razrješavanje obavlja

Napomena

- Razrješavanjem dviju jediničnih klauzula izvodi se prazna klauzula *NIL*

PRIMJER REZOLUCIJE

Nađi rezolventu za sljedeće klauzule

- P(g(y), x, f(z)) ∨ Q(z, b) ∨ R(x)
- S(x, y) ∨ ~P(x, y, f(a))
- Klauzule **nisu standadizirane**! (simboli za varijable x i y pojavljuju se u obje klauzule)

Potrebno je preimenovati varijable u prvoj klauzuli:

x postaje varijabla w,

y postaje varijabla u.

Standardizirane klauzule:

- **(I1) P(g(u), w, f(z)) ∨ Q(z, b) ∨ R(w)**
- **(I2) S(x, y) ∨ ~P(x, y, f(a))**

Nađi mgu!

PRIMJER REZOLUCIJE

- (I1) $P(g(u), y, f(a)) \vee Q(a, b) \vee R(y)$
- (I2) $S(g(u), y) \vee \sim P(g(u), y, f(a))$

- Razrješavanjem po $P(g(u), y, f(a))$ i $\sim P(g(u), y, f(a))$ dobiva se resolventna klauzula:

(I3) $S(g(u), y) \vee Q(a, b) \vee R(y)$

Rezolucijsko pravilo je ispravno (zdravo)

- Prema pravilu univerzalne specijalizacije **I_1δ** je logička posljedica od I_1. Slično, **I_2δ** je logička posljedica od I_2
- Zaključivanjem istim kao i u propozicijskoj logici zaključujemo da je **I_3** je logička posljedica **I_1δ i I_2δ.**
- Dakle **I_3 je logička posljedica I_1 i I_2** te je **rezolucijsko pravilo u predikatnoj logici ispravno**.
- Literali L_1 i M_1 su unificirani pomoću mgu pa je resolventa I_3 u najopćenitijem mogućem obliku

Rezolucija opovrgavanjem je potpuna

- Kao i u propozicijskoj logici tako i u predikatnoj logici vrijedi da je rezolucija opovrgavanjem potpuna. (Robinson, 1965)
- Formula G je logička posljedica premisa F1, F2, .. , Fn akko možemo pokazati da je ulazni skup F1, F2, …, Fn, ~G nekonzistentan izvođenjem resolventne klauzule NIL.
- Svojstvo da je rezolucija zdrava i da je rezolucija opovrgavanjem potpuna osigurava da možemo uvijek dokazati teorem G, ako je G teorem.
- Međutim ako G nije teorem postupak dokazivanja teorema rezolucijom opovrgavanjem može nikad ne završiti – u tom je smislu predikatna logika **poluodlučljiva**

TEMELJNE KLAUZULE I ODLUČLJIVOST

- Literali koji sadrže samo konstante (bez varijabli) nazivaju se **temeljni literali** (*engl. ground literals*), a njihova disjunkcija naziva se **temeljna klauzula** (*engl. ground clause*)

- Podrazred predikatne logike koji dopušta samo pojavljivanje konstanti, funkcija i predikata (a ne varijabli) je **ODLUČLJIV**

- Ta potklasa ima istu moć zaključivanja kao i propozicijska logika

Primjer rezolucije opovrgavanjem u podrazredu predikatne logike koja ne dozvoljava pojavljivanje varijabli.

- Za isti ovaj primjer, cilj "Majka je zadovoljna" dokazali smo u propozicijskoj logici:
  1. prirodnim zaključivanjem i
  2. rezolucijom opovrgavanjem.

[i] Ivan se probudio

[ii] Ivan nosi pribor za čišćenje

[iii] Majka je zadovoljna ako se Ivan probudi i čisti svoju sobu

[iv] Ako Ivan nosi pribor za čišćenje, tada on čisti svoju sobu

- Dokažite rezolucijom opovrgavanjem cilj: *Majka je zadovoljna*

PRIMJER

- Ulazni skup tvore premise i negacija cilja pretvorene u klauzalnu formu

[I1] BUDAN(Ivan)

[I2] NOSI(Ivan, pribor)

[I3] ~BUDAN(Ivan) ∨ ~ČISTI(Ivan, soba) ∨ ZADOVOLJNA(majka)

[I4] ~NOSI(Ivan, pribor) ∨ ČISTI (Ivan, soba)

[I5] ~ZADOVOLJNA(majka) **CILJ**

- Iz ulaznog skupa izvodimo slijedeće resolvente:

[I3] ~BUDAN(Ivan) ∨ ~ČISTI(Ivan, soba) ∨ ZADOVOLJNA(majka)

[I5] ~ZADOVOLJNA(majka)

[I6] ~BUDAN(Ivan) ∨ ~ČISTI(Ivan, soba)

PRIMJER

[I6] ~BUDAN(Ivan) ∨ ~ČISTI(Ivan, soba)

[I1] BUDAN(Ivan)

  [I7] ~ČISTI(Ivan, soba)

[I4] ~NOSI(Ivan, pribor) ∨ ČISTI (Ivan, soba)

  [I8] ~NOSI(Ivan, pribor)

[I2] NOSI(Ivan, pribor)

  [I9] *NIL*

VAŽNOST FAKTORIZIRANJA KLAUZULA U PREDIKATNOJ LOGICI

- Slično kao i u propozicijskoj logici, klauzule u predikatnoj logici moraju biti faktorizirane kako bi se sačuvala potpunost postupka opovrgavanjem

Primjer kako rezolucija gubi svoju potpunost ako nema faktorizacije:

[I1] $P(u) \vee P(w)$

[I2] $\sim P(x) \vee \sim P(y)$

- Uz uporabu mgu = $\{u/x\}$ i razrješavanjem po $P(u)$ i $\sim P(u)$ dobivamo resolventu

[I3] $P(w) \vee \sim P(y)$

(Razrješavanjem po drugim literalima dobivamo alfabetske varijante od I3)

FAKTORIZACIJA U PREDIKATNOJ LOGICI

- Klauzula u predikatnoj logici može biti faktorizirana akko sadrži literale koji se mogu unificirati

[I] $L1 \vee L2 \vee L3 \vee \ldots \vee Ln$

- Pretpostavimo (bez gubitka općenitosti) da se literali L1 i L2 mogu unificirati sa MGU δ. Gornja klauzula tada može biti faktorizirana dajući klauzulu I'

[I'] $L2\delta \vee L3\delta \vee \ldots \vee Ln\delta$

- Klauzula I' se naziva **faktor-klauzula** od I
- Ako skup literala ima MGU δ tada je $I\delta$ faktor klauzula od I, gdje su višestruka pojavljivanja literala u $I\delta$ zamijenjena jednim pojavljivanjem literala

Primjer:

[I1] $P(u) \vee P(w)$

[I2] $\sim P(x) \vee \sim P(y)$

- Uz faktorizaciju:
- Faktoriziramo I1 uporabom mgu={w/u} dobivamo

[I1']   P(w)

- Faktoriziramo I2 uporabom mgu={y/x} dobivamo

[I2']   ~P(y)

- Razrješavanjem jediničnih klauzula P(w) i ~P(y) uz uporabu mgu = {w/y} izvodimo praznu klauzulu

[I3']   NIL

Primjer

$P(x, y, f(b)) \vee S(x, y) \vee P(g(u), w, f(z))$

gdje su: P i S - predikatini simboli,

f i g - funkcijski simboli,

x, y, u, w, z – varijable,

b - konstanta.

- Prvi i treći literal mogu se unificirati pomoću mgu $\delta = \{g(u)/x, y/w, b/z\}$. Faktor klauzula je:

$$P(g(u), y, f(b)) \vee S(g(u), y)$$

FAKTORIZACIJA U PREDIKATNOJ LOGICI

- Faktor koji se sastoji samo od jednog literala naziva se **jedinični faktor**.
- Od sada pa nadalje će se smatrati da je resolventa I3 izvedena iz klauzula I1 i I2 akko dobivamo I3 na bilo koji od sljedećih načina:
  1) razrješavanjem I1 i I2
  2) razrješavanjem I1 i faktora od I2
  3) razrješavanjem faktora od I1 i I2
  4) razrješavajnjem faktora od I1 i faktora od I2

REZOLUCIJSKI POSTUPAK OPOVRGAVANJEM (ALGORITAM)

Da bi se dokazalo da je ciljna formula G deduktivna posljedica od F1, F2, …, Fn, potrebno je primijeniti sljedeće korake:

1. Pretvori F1, F2, F3, …, Fn, ~G u **klauzalni oblik**.
2. Izaberi iz skupa klauzula dvije klauzule koje su razrješive. Ako je potrebno, standardiziraj te klauzule. **Izvedi resolventu i dodaj je skupu klauzula**. Ponavljaj ovaj korak sve dok se ne ispuni jedan od uvjeta:
   - Izvedena je prazna klauzula *NIL*. (Dokazano je da je cilj G teorem, ulazni skup je nekonzistentan.)
   - Niti jedan par klauzula ne može se razriješiti ili ne može se izvesti niti jedna nova klauzula (Dokazano je da cilj G nije teorem)
   - Neki unaprijed zadani resursi računala su iscrpljeni. (Nema odluke o G – predikatna logika je poluodlučljiva)

Primjer

[1] $\forall x(\text{MUŽ}(x) \rightarrow \text{VOLI}(x, \text{žena}(x)))$

[2] MUŽ(Marko)

Dokaži uporabom rezolucije opovrgavanjem:

VOLI(Marko, žena(Marko)).

Premise i negacija ciljne formule pretvore se u klauzalni oblik

[I1] $\sim\text{MUŽ}(x) \vee \text{VOLI}(x, \text{žena}(x))$

[I2] MUŽ(Marko)

[I3] ~VOLI(Marko, žena(Marko))

Iz ulaznog skupa rezolucijom izvodimo:

[I4] ~MUŽ(Marko) - uporabom **mgu**={Marko/x} iz [I1] i [I3].

[I5] *NIL* - uporabom {} iz [I2] i [I4].

Time je dokazano da je VOLI(Marko, žena(Marko)) dedukcija

PRIMJER

Primjer

- Robot dostavlja pakete.
- Robot zna da su svi paketi u sobi 27 manji od svakog paketa u sobi 28.
- A i B su paketi.
- Paket A je u sobi 27 ili u sobi 28, ali robot ne zna u kojoj.
- Paket B je u sobi 27 i nije manji od paketa A.

Uporabom rezolucije opovrgavanjem pokaži kako robot može zaključiti da je paket A u sobi 27.

- **KONSTANTE**

  27, 28, A, B
- **PREDIKATI**

  PAKET(x) - x je paket – skraćeno P(x)

  U_SOBI(x, y) - x je u sobi y – skraćeno U(x, y)

  MANJI(x, y) - x je manji od y – M(x,y)

PRIMJER

- **BAZA ZNANJA**

$\forall x\ \forall y\ \big(\big(\ P(x) \wedge P(y) \wedge U(x, 27) \wedge U(y, 28)\ \big) \rightarrow M(x, y)\big)$

$P(A)$

$P(B)$

$U(A, 27) \vee U(A, 28)$

$U(B, 27) \wedge \sim M(B, A)$

PRIMJER

- **BAZA ZNANJA**

  $\forall x \, \forall y \, (P(x) \wedge P(y) \wedge U(x, 27) \wedge U(y, 28)) \rightarrow M(x, y)$

  $P(A)$

  $P(B)$

  $U(A, 27) \vee U(A, 28)$

  $U(B, 27) \wedge \sim M(B, A)$

- **BAZA ZNANJA u klauzalnoj formi**

  $\sim P(x) \vee \sim P(y) \vee \sim U(x, 27) \vee \sim U(y, 28) \vee M(x, y)$

  $P(A)$

  $P(B)$

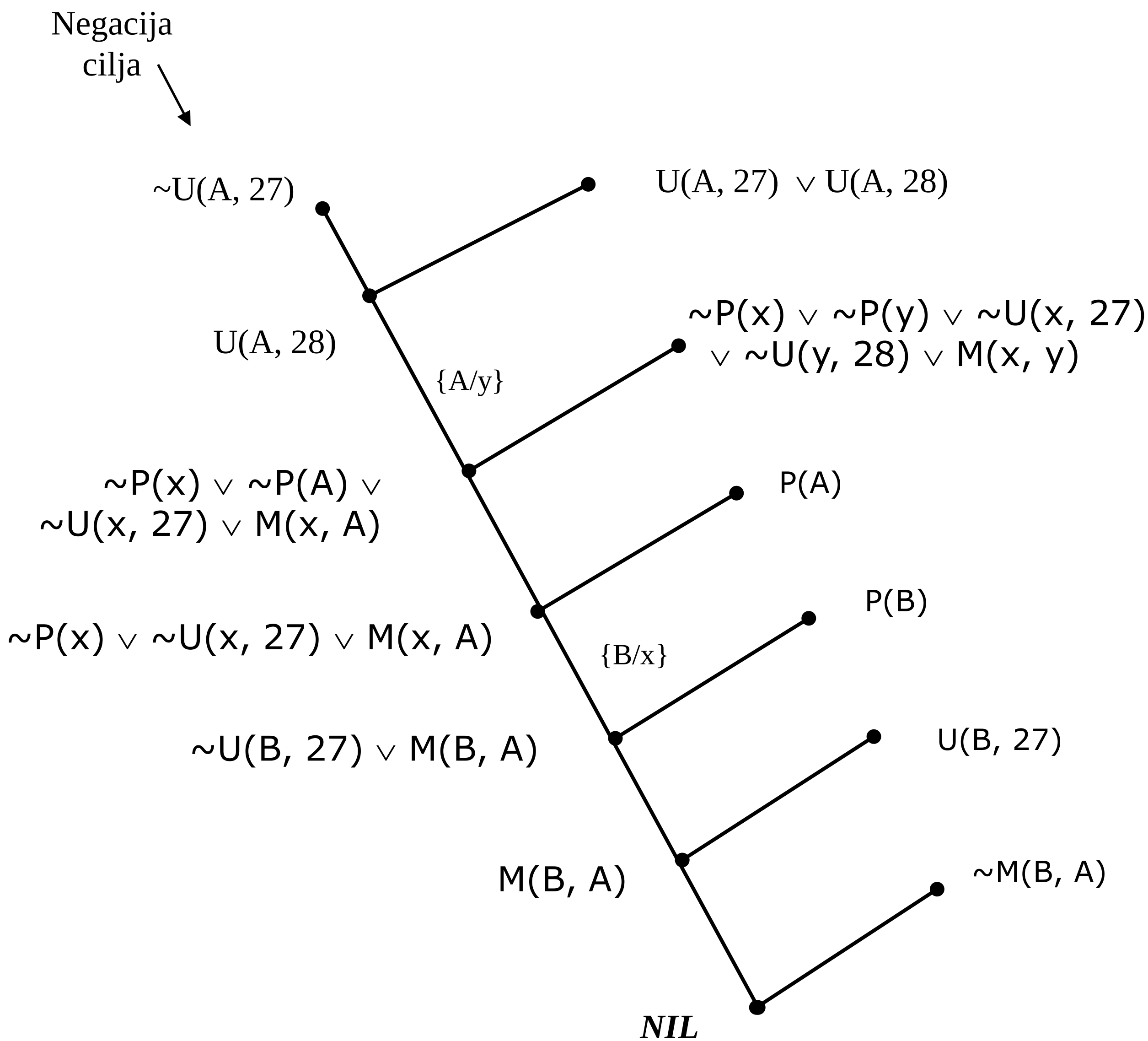
  $U(A, 27) \vee U(A, 28)$

  $U(B, 27)$

  $\sim M(B, A)$

  $\sim U(A, 27)$ (negacija cilja)

STABLO DOKAZA



- **Binarna rezolucija** je kombiniranje dvije klauzule koje sadrže komplementarne literale

Primjer

$Q(x) \vee \sim P(x, a)$

$\sim Q(b) \vee R(x)$

Uz unifikaciju {b/x} resolventna klauzula je

$\sim P(b, a) \vee R(b)$

- **Rezolucija s rezultirajućom jediničnom klauzulom** (engl. *unit resulting resolution*)

  Istodobno razrješavanje više klauzula kako bi se izvela jedinična klauzula. Sve roditeljske klauzule osim jedne su jedinične, a ta ima točno jedan literal više od ukupnog broja jediničnih klauzula

Primjer

VJENČANI(Ana, Marko)

~OTAC(Marko, Ivan)

~VJENČANI(x, y) ∨ ~MAJKA(x, z) ∨ OTAC(y, z)

(uz supstituciju {Ana/x, Marko/y, Ivan/z} resolventa je jedinična klauzula)

~MAJKA(Ana, Ivan)

- **Linearna rezolucija**

  Ako je uvijek jedna od roditeljskih klauzula izvedena u prethodnom koraku

- **Linearna rezolucija na ulaznom skupu**

  Ako je jedna od roditeljskih klauzula uvijek iz izvornog ulaznog skupa klauzula

STRATEGIJE REZOLUCIJE

Unifikacija + rezolucija → **automatsko zaključivanje**

ALI

- Rezolucija je nedjelotvorna bez daljnjih razrada!
- Razrješavanje slučajno odabranih klauzula → kombinatorna eksplozija
- Važna uporaba metoda koje ograničavaju pretraživanje

Odabir redoslijeda razrješavanja klauzula kako bi **postupak rezolucije bio djelotvorniji** zove se **STRATEGIJA**

STRATEGIJE REZOLUCIJE

1. **Uređajne strategije** (engl. *ordering strategies*) – određuju slijed (poredak) kojim će se klauzule razrješavati
2. **Strategije skraćivanja** (engl. *pruning strategies*) – uklanjanje iz skupa klauzula onih klauzula i literala koji nisu neophodni za dokaz
3. **Strategije ograničavanja** (engl. *restriction strategies*) – propisuju primjenu rezolucije samo na one klauzule koje se smatraju vitalne za dokaz

STRATEGIJA SKUPA POTPORE

- Jedna od najvažnijih strategija (iz skupa strategija ograničavanja)
- Neka je S kontradiktoran skup klauzula i neka je T podskup od S. Tada je T **skup potpore** od S ako je S\T konzistentan.
- Rezolucija skupa potpore je rezolucija u kojoj nikad nisu obje klauzule iz S\T. Za svaku rezolventu barem je jedna roditeljska klauzula iz T.
  - pretpostavka: baza znanja je konzistentna, jer inače nam ni dokazivanje cilja ne znači puno
  - dodatna prednost – stabla dokaza su razumljiva jer su usmjerena cilju

EKSTRAKCIJA ODGOVORA POMOĆU REZOLUCIJE OPOVRGAVANJEM

- Ako je neka formula oblika **∃xG(x)** logička posljedica nekih premisa tada to možemo dokazati rezolucijom opovrgavanjem.

- Rezolucijom opovrgavanjem možemo i više od toga tj. odgovoriti na pitanje:

Za koje vrijednosti x *je G(x) logička posljedica premisa?*

EKSTRAKCIJA ODGOVORA POMOĆU REZOLUCIJE OPOVRGAVANJEM

Primjer:

- Pas Fido je svagdje gdje je njegov gazda Ivan.
- Ivan je u knjižnici.

Gdje je Fido?

Aksiomi:

$\forall x$ JE_U(Ivan, x) $\rightarrow$ JE_U(Fido, x)

JE_U(Ivan, knjižnica)

Klauzalna forma:

~JE_U(Ivan, x) $\vee$ JE_U(Fido, x)

JE_U(Ivan, knjižnica)

Hipoteza:

$\exists z$ JE_U(Fido, z) (z?)

Negacija hipoteze:

$\forall z$ ~JE_U(Fido, z) …… (Fido je nigdje)

~JE_U(Fido, z) (klauzalna forma negacije hipoteze)

EKSTRAKCIJA ODGOVORA POMOĆU REZOLUCIJE OPOVRGAVANJEM

~JE_U(Fido, z)   ~JE_U(Ivan, x) $\vee$ JE_U(Fido, x)

{x/z}

~JE_U(Ivan, x)   JE_U(Ivan, knjižnica)

{knjižnica/x}

NIL

EKSTRAKCIJA ODGOVORA

- Supstitucije pod kojima je nađena kontradikcija jesu supstitucije pod kojima je pretpostavka (hipoteza) istinita.
- Čuvanje informacije o unifikacijama tijekom dokaza omogućava odgovor na upit

Odgovor:

hipoteza	**supstitucije**	**odgovor na pitanje**
$\exists$z JE_U(Fido, z)	{x/z, knjižnica/x} =	**JE_U(Fido, knjižnica)**

Napomena: Nađemo li kompoziciju ovih dviju suspstitucija:

α ={x/z} i β ={knjižnica/x} (postupkom u dva koraka) dobit ćemo α β = {knjižnica/z, knjižnica/x}

Primjer

1. Ankica je mama od Branke	MAJKA(Ankica, Branka)
2. Za sve x i y vrijedi: *Ako je x kćerka od y onda* *je y majka od x*	$\forall x \forall y$ (KĆERKA(x, y) $\rightarrow$ MAJKA(y, x))
3. Zorica je kćerka od Branke.	KĆERKA (Zorica, Branka)
4. Za sve x, y, z vrijedi: *Ako je x majka od y, i y majka od z,* *tada je x baka od z.*	$\forall x \forall y \forall z$(MAJKA(x, y) $\wedge$ MAJKA(y, z)) $\rightarrow$ BAKA(x, z))

Dokaži rezolucijom opovrgavanjem: Zorica ima baku tj.

∃v(BAKA(v, Zorica))

Premise u klauzalnom obliku + negacija cilja:

[I1] MAJKA(Ankica, Branka)

[I2] ~KĆERKA(u, w) ∨ MAJKA(w, u)

[I3] KĆERKA (Zorica, Branka)

[I4] ~MAJKA(x, y) ∨ ~MAJKA(y, z) ∨ BAKA(x, z)

[I5] ~BAKA(v, Zorica)

Izvodimo

[I6] ~MAJKA(x, y) ∨ ~MAJKA(y, Zorica)	mgu={x/v, Zorica/z}, [I4] i [I5]
[I7] ~KĆERKA(Zorica, y) ∨ ~MAJKA(x, y)	mgu={y/w, Zorica/u}, [I2] i [I6]
[I8] ~MAJKA (x, Branka)	mgu={Branka/y} [I3] i [I7]
[I9] **NIL**	mgu={Ankica/x} [I1] i [I8]

PRIMJER

- Uporaba rezolucije opovrgavanjem za odgovor na pitanje iz skupa premisa [I1]-[I4]

Tko je Zoričina baka?

- Ne zanima nas samo odgovor postoji li *v* tako da je *v* baka od Zorice tj. ∃v (BAKA(v, Zorica)), nego nas zanima vrijednost varijable *v*

Postupak:

- Svaka klauzula koja se dobije kao negacija cilja pretvara se u tautologiju. To se radi tako da klauzuli dodajemo negaciju svakog literala koji ona sadrži. (Ako je G(x) negacija cilja, tada je G(x) ∨ ~G(x) tautologija)

PRIMJER

[J1] MAJKA(Ankica, Branka)

[J2] ~KĆERKA(u, w) ∨ MAJKA(w, u)

[J3] KĆERKA (Zorica, Branka)

[J4] ~MAJKA(x, y) ∨ ~MAJKA(y, z) ∨ BAKA(x, z)

[J5] ~BAKA(v, Zorica) ∨ BAKA(v, Zorica)

- Sada ponavljamo potpuno isti postupak kao u dokazivanju formule ∃v(BAKA(v, Zorica)).

- Rezultat: umjesto *NIL* – odgovor na pitanje *tko je Zoričina baka*

PRIMJER

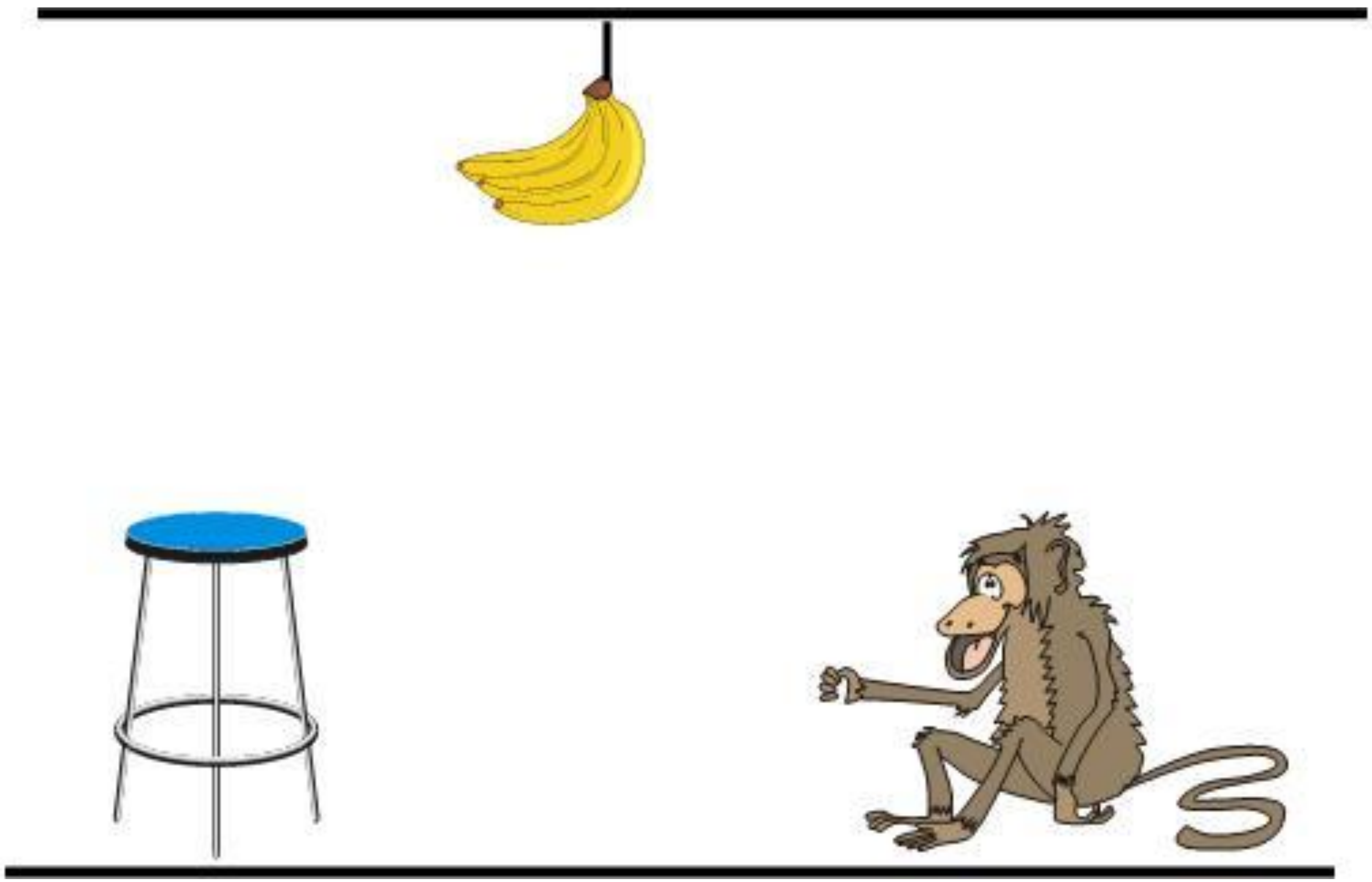
1. ~MAJKA(x, y) ∨ ~MAJKA(y, Zorica) ∨ BAKA(x, Zorica)	mgu={x/v, Zorica/z}, [J4] i [J5]
2. ~KĆERKA(Zorica, y) ∨ ~MAJKA(x, y) ∨ BAKA(x, Zorica)	mgu={y/w, Zorica/u}, [J2] i [J6]
3. ~MAJKA (x, Branka) ∨ BAKA(x, Zorica)	mgu={Branka/y} [J3] i [J7]
4. BAKA(Ankica, Zorica)	mgu={Ankica/x} [J1] i [J8]

Ankica je Zoričina baka

Primjer: Majmun i banane

PRIMJER REZOLUCIJE

- Majmun i banane



PRIMJER REZOLUCIJE

U sobi se nalaze:

- majmun,
- stolica,
- banane koje vise sa sredine stropa, ali na visini koja nije na dohvat ruke majmunu.

Ako je majmun dovoljno bistar, on može:

- postaviti stolicu ispod snopa banana,
- popesti se na stolicu i
- dohvatiti banane.

Zadatak:

1. Uporabom FOPL prikaži činjenice iz svijeta “majmun-banane”
2. Uporabom rezolucije opovrgavanjem dokaži da majmun može dohvatiti banane

PRIMJER REZOLUCIJE

- U kreiranju baze znanja važno je odrediti:
  - sve relevantne objekte (majmun, banane, stolica, pod)
  - odnose među njima (npr. banane nisu blizu poda, stolica se može pomaknuti ispod banana).
- Sve nevažno treba izostaviti (npr. prozori, vrata…)

KONSTANTE

pod, stolica, banana, majmun

VARIJABLE

x, y, z

PREDIKATI

MOŽE_DOHVATITI(x, y)	*x može dohvatiti y*
SPRETAN(x)	*x je spretan*
BLIZU(x, y)	*x je blizu y*
JE_NA(x, y)	*x je na y*

PRIMJER REZOLUCIJE

ISPOD(x, y)	*x je ispod y*
VISOKA(x)	*x je visok(a)*
U_SOBI(x)	*x je u sobi*
MOŽE_POMAKNUTI_BLIZU(x, y, z)	*x može pomaknuti y blizu z*
MOŽE_POPETI_NA(x, y)	*x se može popeti na y*

AKSIOMI BAZE ZNANJA

U_SOBI(banane)
U_SOBI(stolica)
U_SOBI(majmun)
VISOKA(stolica)
SPRETAN(majmun)
MOŽE_POMAKNUTI_BLIZU(majmun, stolica, banane)
MOŽE_POPETI_NA(majmun, stolica)
~BLIZU(banane, pod)

PRIMJER REZOLUCIJE

$\mathbf{(}$U_SOBI(x) $\wedge$ U_SOBI(y) $\wedge$ U_SOBI(z) $\wedge$ MOŽE_POMAKNUTI_BLIZU(x, y, z)$\mathbf{)}$ $\rightarrow$ BLIZU(z, pod) $\vee$ ISPOD(y, z)

MOŽE_POPETI_NA(x, y) $\rightarrow$ JE_NA(x, y)

$\mathbf{(}$JE_NA(x, y) $\wedge$ ISPOD(y, banane) $\wedge$ VISOK(y)$\mathbf{)}$ $\rightarrow$ BLIZU(x, banane)

$\mathbf{(}$SPRETAN(x) $\wedge$ BLIZU(x, y)$\mathbf{)}$ $\rightarrow$ MOŽE_DOHVATITI(x, y)

- **Aksiomi se mogu pretvoriti u klauzalnu formu (uporabom De Morganovih zakona i ekvivalencije**
  $P \rightarrow Q \equiv \sim P \vee Q$)

PRIMJER REZOLUCIJE

Klauzalni oblik baze znanja

1. U_SOBI(banane)
2. U_SOBI(stolica)
3. U_SOBI(majmun)
4. VISOKA(stolica)
5. SPRETAN(majmun)
6. MOŽE_POMAKNUTI_BLIZU(majmun, stolica, banane)
7. MOŽE_POPETI_NA(majmun, stolica)
8. ~BLIZU(banane, pod)
9. ~MOŽE_POPETI_NA(x, y) ∨ JE_NA(x, y)
10. ~SPRETAN(x) ∨ ~BLIZU(x, y) ∨ MOŽE_DOHVATITI(x, y)
11. ~JE_NA(x, y) ∨ ~ISPOD(y, banane) ∨ ~VISOK(y) ∨ BLIZU(x, banane)
12. ~U_SOBI(x) ∨ ~U_SOBI(y) ∨ ~U_SOBI(z) ∨ ~MOŽE_POMAKNUTI_BLIZU(x, y, z) ∨ BLIZU(z, pod) ∨ ISPOD(y, z)
13. ~ MOŽE_DOHVATITI(majmun, banane) (negacija cilja)

Dokaz rezolucijom

14. ~MOŽE_POMAKNUTI_BLIZU(majmun, stolica, banane) ∨ BLIZU(banane, pod) ∨ ISPOD(stolica, banane)

    (resolventa od [1], [2], [3] i [12] uz supstituciju {majmun/x, stolica/y, banane/z})

15. BLIZU(banane, pod) ∨ ISPOD(stolica, banane)

    (resolventa od [6], [14])

16. ISPOD(stolica, banane) *([8] i [15])*
17. ~JE_NA(x, stolica) ∨ ~VISOK(stolica) ∨ BLIZU(x,banane)

    (supstitucija {stolica/y} i [16] i [11])

18. ~JE_NA(x, stolica) ∨ BLIZU(x, banane) *([4] i [17])*

PRIMJER REZOLUCIJE

19. ~JE_NA(majmun, stolica) *([7] i [9])*
20. BLIZU(majmun, banane) *([18] i [19] supstitucija {majmun/x})*
21. ~BLIZU(majmun, y) $\vee$ MOŽE_DOHVATITI(majmun, y) *([10] i [5] supstitucija {majmun/x})*
22. MOŽE_DOHVATITI(majmun, y) *([20] i [21] supstitucija {banane/y})*
23. **{}** *([13] i [22])*

PRIMJER REZOLUCIJE

- Ovaj dokaz nije vođen niti jednom posebnom strategijom iako se vodilo računa o odabiru roditeljskih klauzula. U suprotnom, puno nepotrebnih koraka može biti učinjeno.

Zadatak

- *Možete li komentirati koji su svi oblici rezolucije učinjeni kroz korake od [14] do [23]?*

Primjer: Financijski savjetnik

CASE STUDY - Financijski savjetnik

- Uporaba predikatne logike u predstavljanju znanja i zaključivanju iz problemske domene – financijsko savjetovanje
- Pomoć u odluci da li sredstva:
  - pohraniti na bankovni račun (štednja)
  - ulagati u dionice
  - kombinirati gornje
- Preporuke su individualne **i zavise od svakog pojedinog ulagača tj. od:**

1. **uštede** ulagača – koja zavisi o stanju na bankovnom računu pojedninca i broj uzdržavanih članova obitelji.
2. **prihoda** – iznos prihoda, broj uzdržavanih članova obitelji, stalnost prihoda

CASE STUDY - Financijski savjetnik

Preporuke:

1. Pojedinci sa ***neodgovarajućom uštedom*** trebaju uvijek prvo povećati uštedu, bez obzira na prihod.
2. Pojedinci sa ***odgovarajućom uštedom*** i ***odgovarajućim prihodima*** trebaju razmotriti profitabilnija i rizičnija ulaganja u dionice.
3. Pojedinci s ***neodgovarajućim prihodima*** i ***odgovarajućom uštedom*** trebali bi razdijeliti svoja ulaganja između:
   - štednje (povećaju svoju sigurnost) i
   - ulaganja u dionice (pokušaju povećati prihode)

CASE STUDY - Financijski savjetnik

- ***Odgovarajuća ušteda*** – barem 5 000 $ po uzdržavanom članu obitelji.
- ***Odgovarajući prihod*** – barem 15 000 $ godišnje plus 4000$ po uzdržavanom članu obitelji.

Kako modelirati ova znanja u predikatnoj logici?