Ekspertski Sistemi
Vežbe

Formalna logika

Maj 2013.

Program vežbi

- Algoritmi pretraživanja
- Teorija igara
- Formalna logika (model predstavljanja znanja)
- Produkcioni sistemi
- Strategije rešavanja problema
- Uvod u mašinsko učenje
- Rad u neizvesnom okruženju

Zadatak 1: predikati START, END, DUR

- Dati su predikati:
  - START(e,t) – istinit ako je dogadjaj e započeo u trenutku t
  - END(e,t) – istinit ako se neki dogadjaj e završio u trenutku t
  - DUR(e,d) – istinit ako je dogadjaj e trajao d vremenskih jedinica

a)

- Napisati dobro formiranu formulu koja bi omogućila zaključivanje o početku nekog dogadjaja bez obzira da li je poznata vrednost t u predikatu START
- Napisati dobro formiranu formulu koja bi omogućila zaključivanje o kraju nekog dogadjaja, bez obzira na to da li je poznata vrednost t u predikatu END

Prvi način

- Za zaključivanje o početku dogadjaja

¥e ¥t ¥d

{ [END(e,t) ∧ DUR(e,d)] => START(e,t-d) }

- Za zaključivanje o kraju dogadjaja

¥e ¥t ¥d

{ [START(e,t) ∧ DUR(e,d)] => END(e,t+d) }

Drugi način

- Za zaključivanje o početku dogadjaja

¥e ¥t ¥d

{ [END(e,t) ∧ DUR(e,d) ∧ RAZ(r,t,d)] => START(e,r) }

- Za zaključivanje o kraju dogadjaja

¥e ¥t ¥d

{ [START(e,t) ∧ DUR(e,d) ∧ ZBIR(z,t,d)] => END(e,z) }

b) Definisati novi predikat AFTER(e2,e1) koji je istinit ako se dogadjaj e2 desio posle dogadjaja e1, koristeći formule iz prethodne tačke

AFTER(e2,e1)

- Potrebno je da bude ispunjeno:
  - Vremenski trenutak t2 (kada je počeo dogadjaj e2) sledi vremenski trenutak t1 (kada se završava dogadjaj e1)

AFTER(e2,e1)

¥e1 ¥e2 ¥t1 ¥t2

[START(e2,t2) ∧

END(e1,t1) ∧

GT(t2, t1) => AFTER(e2, e1)]

- Predikat GT(t2, t1) je tačan ako je t2 veće od t1.

c) Definisati novi predikat IN(e1,e2) koji je istinit ako se dogadjaj e1 završio u trenutku kada se e2 dešava

IN(e1,e2)

- Potrebno je da bude ispunjeno
  - Vremenski trenutak t1 kada se završava dogadjaj e1 treba da se nalazi izmedju vremena početka t2 i kraja t3 dogadjaja e2

IN(e1,e2)

¥e1 ¥e2 ¥t1 ¥t2 ¥t3

[END(e1,t1) ∧

START(e2,t2) ∧

END(e2,t3) ∧

BET(t1 , t2 , t3) => IN(e1, e2)]

BET ?

Zadatak 2: Ostrvo uživanja

- Čokoladgrad, glavni grad Ostrva uživanja, nalazi se na raskrsnici četiri puta koja postoje na ostrvu. Svaki od puteva vodi na jednu od četiri strane sveta i ide do jednog od četiri sela

```
            selo
             ||
selo ==== Č.grad  ==== selo
             ||
            selo
```

- Poznato je sledeće:
  - Selo na Okeanskom putu za 1 milju je dalje od glavnog grada nego Sladoled selo
  - Najbliže selo udaljeno je 2 milje od Čokoladgrada
  - Ne postoje sela koja su podjednako udaljena od Čokoladgrada
  - Selo Urmašica udaljeno je 6 milja od sela koje se nalazi na Obalskom putu
  - Selo na Putu Slasti udaljeno je 9 milja od Slatkog sela
  - Selo na jugu dva puta je dalje od glavnog grada nego selo na severu

- Predstaviti ove činjenice u obliku stavova predikatske logike

```
                 selo
                  ||
selo ==== Č.grad  ==== selo
                  ||
                 selo
```

- U cilju rešenja usvajamo sledeće predikate
  - NA_PUTU(p,s) je ispunjeno ako je selo s na putu p
  - U_SMERU(d,s) je ispunjeno ako je selo s na strani sveta d
  - JEDNAKO(x,y) označava relacioni operator jednakosti x i y, gde x i y mogu biti oznake sela ili razdaljine (overloading operatora)
  - MANJE(x,y) je ispunjeno ako je razdaljina x manja od razdaljine y
  - DALJINA(x,y) funkcija koja kao rezultat vraća udaljenost x od y

- Selo na Okeanskom putu za 1 milju je dalje od glavnog grada nego Sladoled selo

¥x [
  NA_PUTU(Okeanski_put,x) =>
  JEDNAKO(
    DALJINA(x,Čokoladgrad),
    DALJINA(Sladoled_Selo,Čokoladgrad)+1
)
]

- Najbliže selo udaljeno je 2 milje od glavnog grada

$\neg \exists x \, [\, \mathrm{MANJE}(\mathrm{DALJINA}(x, \text{Čokoladgrad}), 2) \,]$

- Ne postoje sela koja su podjednako udaljena od glavnog grada

```
¥x ¥y [
  JEDNAKO(
    DALJINA(x, Čokoladgrad),
    DALJINA(y, Čokoladgrad)
  ) => JEDNAKO(x,y)
]
```

- Selo Urmašica udaljeno je 6 milja od sela koje se nalazi na Obalskom putu

¥x [

NA_PUTU(Obalski_put,x)

=> JEDNAKO(DALJINA(x, Urmašica), 6)

]

- Selo na Putu Slasti udaljeno je 9 milja od Slatkog sela

¥x [

NA_PUTU(Put_slasti,x)

=> JEDNAKO(DALJINA(x, SlatkoSelo), 9)

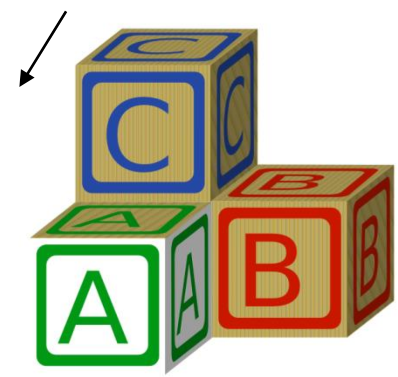
]

- Selo na jugu dva puta je dalje od glavnog grada nego selo na severu

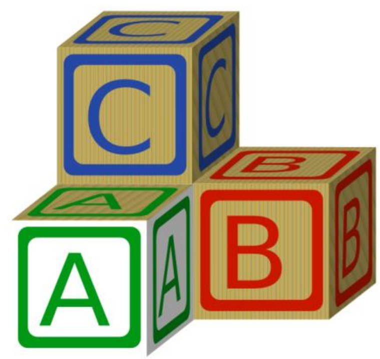
¥x ¥y [
  U_SMERU(Jug, x) ∧
  U_SMERU(Sever,y)
  =>
  JEDNAKO(
    DALJINA(x, Čokoladgrad),
    2*DALJINA(y, Čokoladgrad)
)
]

Zadatak 3: Svet blokova (interpretacija predikatskih formula)

- Svet blokova predstavlja jednu od interpretacija sledećih dobro formiranih formula
  - ON(C,A)
  - ONTABLE(A)
  - ONTABLE(B)
  - CLEAR(C)
  - CLEAR(B)
  - (¥x) [CLEAR(x) => ¬(Эy) ON(y,x)]



- Naći najmanje dve drugačije interpretacije (van sveta blokova)

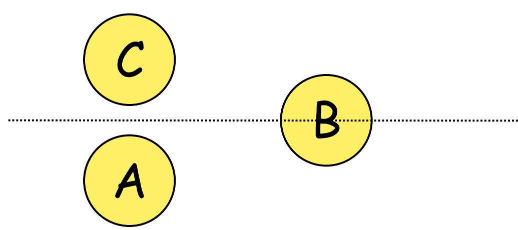


Interpretacija predikatskih formula

- Dodela značenja (semantike) pojedinim predikatima, funkcijama, konstantama i promenljivama
- Imena ne moraju da odslikavaju interpretaciju
- Možemo odrediti proizvoljan broj različitih interpretacija za istu formulu

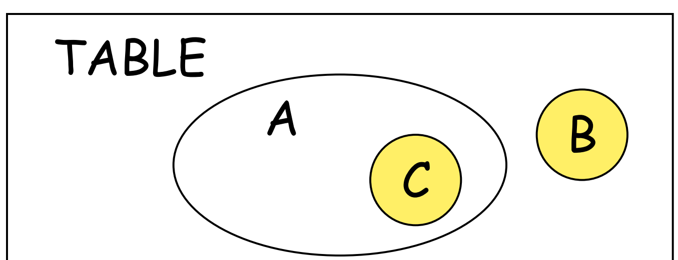
Interpretacija odnosa zaposlenih u preduzeću

- *A*, B, *C* su osobe koje rade u istom preduzeću
- Predikat ON(*C*,*A*) označava da je osoba *C* nadredjena osobi *A*
- Predikati ONTABLE(*A*) i ONTABLE(B) označavaju da su osobe *A* i B izvršioci poslova
- Predikati CLEAR(*C*) i CLEAR(B) označavaju da su B i *C* vlasnici firme
- Formula (¥x) [CLEAR(x) => ¬(Эy) ON(y,x)] označava da vlasnik preduzeća nema nadredjenu osobu



Interpretacija matematičkog pojma skupa

- A, B, C predstavljaju različite skupove
- Predikat ON(C,A) označava da je C podskup skupa A
- Predikati ONTABLE(A) i ONTABLE(B) označavaju da su skupovi A i B podskupovi skupa TABLE
- Predikati CLEAR(C) i CLEAR(B) označavaju da su B i C prazni skupovi
- Formula (¥x) [CLEAR(x) => ¬(Эy) ON(y,x)] opisuje da prazan skup ne može imati podskup



Zadatak 4: Jovanovi preci

- Pretpostavimo da smo činjenicu da je Jovan Petrov otac predstavili sa OTAC(Petar, Jovan)
- Činjenicu da je Milica Petrova majka predikatom MAJKA(Petar, Milica)
- Činjenicu da je Milan jedan od Petrovih predaka predikatom PREDAK(Petar, Milan)

- Napisati dobro formiranu formulu koja treba da predstavi činjenicu:

  Svaki Petrov predak je ili njegov otac ili njegova majka ili jedan od njihovih predaka

¥x {

PREDAK(Petar,x) =>

[

OTAC(Petar,x) v MAJKA(Petar,x) v

[Эy(OTAC(Petar,y) v MAJKA(Petar,y)) λ PREDAK(y,x)]

]

}

- Preciznije, odnos je XOR:

  ili njegov otac ili njegova majka ili jedan od njihovih predaka

- $P \text{ xor } Q = (P \wedge \neg Q) \vee (\neg P \wedge Q)$

  $= (P \vee Q) \wedge (\neg P \vee \neg Q)$

  $= (P \vee Q) \wedge \neg(P \wedge Q)$

Zadatak 5: Nalaženje konjuktivne normalne forme

Odrediti konjuktivnu normalnu formu za sledeću formulu:

¥x { Cigla(x) =>
  {
    Эy [Na(x,y) ∧ ¬Piramida(y)] ∧
    ¬Эy [Na(x,y) ∧ Na(y,x)] ∧
    ¥y [¬Cigla(y) => ¬Jednako(x,y)]
  }
}

- Svaka dobro formirana formula može se dovesti u konjuktivnu normalnu formu
- Niz klauzula povezanih konjukcijom
- Klauzula je niz literala povezanih disjunkcijom

$(a \lor b \lor c) \land (....)$ ....

Korak 1: Eliminisanje implikacije

- (E1 => E2 transformiše se u ¬E1 v E2)

Korak 1: Eliminisanje implikacije

¥x { ¬ Cigla(x) v

{

Эy [Na(x,y) ∧ ¬Piramida(y)] ∧

¬Эy [Na(x,y) ∧ Na(y,x)] ∧

¥y [¬(¬Cigla(y)) v ¬Jednako(x,y)]

}

}

Korak 2: “spuštanje” negacija do atomskih formula

- ¬(E1 ∧ E2) transformiše se u ¬E1 ∨ ¬E2
- ¬(E1 ∨ E2) transformiše se u ¬E1 ∧ ¬E2
- ¬(¬E1) transformiše se u E1
- ¬¥x [E1(x)] transformiše se u ∃x [¬E1(x)]
- ¬∃x [E1(x)] transformiše se u ¥x [¬E1(x)]

Korak 2: “spuštanje” negacija do atomskih formula

∀x { ¬ Cigla(x) v

{

∃y [Na(x,y) ∧ ¬Piramida(y)] ∧

∀y [¬Na(x,y) v ¬Na(y,x)] ∧

∀y [Cigla(y) v ¬Jednako(x,y)]

}

}

Korak 3: Uklanjanje egzistencijalnih kvantifikatora

- Posmatrajmo izraz

¥x Эy [Na(x,y) ʌ ¬Piramida(y)]

Korak 3: Uklanjanje egzistencijalnih kvantifikatora

- Postoji funkcija Φ koja za svaku vrednost x daje odgovarajuće y
- Na(x,Φ(x)) ∧ ¬Piramida(Φ(x))
- Generalno, argumenti funkcije su sve promenljive koje su vezane univerzalnim kvantifikatorom na onom mestu u formuli na kome se pojavljuje član y

Korak 3: Uklanjanje egzistencijalnih kvantifikatora

¥x { ¬ Cigla(x) v

{

[Na(x,Drzi(x)) ∧ ¬Piramida(Drzi(x))] ∧

¥y [¬Na(x,y) v ¬Na(y,x)] ∧

¥y [Cigla(y) v ¬Jednako(x,y)]

}

}

Korak 4: Preimenovanje promenljivih (1 kvantifikator – 1 promenljiva)

¥x { ¬ Cigla(x) v

{

[Na(x,Drzi(x)) ∧ ¬Piramida(Drzi(x))] ∧

¥y [¬Na(x,y) v ¬Na(y,x)] ∧

¥z [Cigla(z) v ¬Jednako(x,z)]

}

}

Korak 5: Premeštanje univerzalnih kvantifikatora levo bez promene redosleda

¥x ¥y ¥z { ¬ Cigla(x) v

{

[Na(x,Drzi(x)) ∧ ¬Piramida(Drzi(x))] ∧

[¬Na(x,y) v ¬Na(y,x)] ∧

[Cigla(z) v ¬Jednako(x,z)]

}

}

Korak 6: Spuštanje disjunkcija do najnižeg nivoa

- (E1 ∧ E2) ∨ E3

transformiše se u

(E1 ∨ E3) ∧ (E2 ∨ E3)

Spuštanje disjunkcija do najnižeg nivoa

¥x ¥y ¥z { (¬ Cigla(x) v Na(x,Drzi(x))) ∧

(¬ Cigla(x) v ¬Piramida(Drzi(x)) ∧

(¬ Cigla(x) v ¬Na(x,y) v ¬Na(y,x)) ∧

(¬ Cigla(x) v Cigla(z) v ¬Jednako(x,z))

}

Korak 7: Eliminacija konjukcija

¥x (¬ Cigla(x) v Na(x,Drzi(x)))

¥x (¬ Cigla(x) v ¬Piramida(Drzi(x))

¥x ¥y (¬ Cigla(x) v ¬Na(x,y) v ¬Na(y,x))

¥x ¥z (¬ Cigla(x) v Cigla(z) v ¬Jednako(x,z))

Korak 8: Preimenovanje promenljivih

¥x (¬ Cigla(x) v Na(x,Drzi(x)))

¥u (¬ Cigla(u) v ¬Piramida(Drzi(u))

¥v ¥y (¬ Cigla(v) v ¬Na(v,y) v ¬Na(y,v))

¥w ¥z (¬ Cigla(w) v Cigla(z) v ¬Jednako(w,z))

Korak 9: Uklanjanje kvantifikatora

¬ Cigla(x) v Na(x,Drzi(x))

¬ Cigla(u) v ¬Piramida(Drzi(u))

¬ Cigla(v) v ¬Na(v,y) v ¬Na(y,v)

¬ Cigla(w) v Cigla(z) v ¬Jednako(w,z)

Zaključivanje

Aksiome i teoreme

- Formule koje usvajamo da su istinite nazivamo aksiomama
- Formula koja se dokazuje uz pomoć aksioma naziva se teorema

Zaključivanje

- Generisati nove WFF iz postojećih izraza, tako da ukoliko su stari TRUE i novo dobijeni budu TRUE

Modus Ponens

- Ako postoji aksioma E1 i E1 -> E2, tada E2 logički sledi

Rezolucija

- Ako postoji aksioma E1 v E2
- I postoji aksioma ~E2 v E3
- Sledi da je: E1 v E3

Dokazivanje opovrgavanjem

- Pretpostaviti da važi negacija teoreme (dodati je aksiomama)
- Zaključiti da pretpostavljena negacija vodi ka kontradikciji
- Zaključiti da je teorema istinita jer negacija teoreme ne može biti

Zadatak 6: Saša i kikiriki

Dati su iskazi:

1. Saša voli sve vrste hrane
2. Jabuke su hrana
3. Piletina je hrana

4. Hrana je sve ono što neko jede i ne otruje se
5. Srdjan jede kikiriki i još je živ
6. Ceca jede sve što Srdjan jede

a) Pretvoriti iskaze u dobro formirane formule predikatske logike

- Definisati predikate
- Predikatima predstavljamo osobine objekata. HRANA(x) istinit ako je objekat x vrsta hrane.
- Konkretne objekte predstavljamo konstantama, JABUKA, PILETINA, KIKIRIKI, SRDJAN, CECA
- Predikatima iskazujemo i relacije: VOLI(x, y), OTRUJE_SE(x,y), JEDE(x,y)

1. Saša voli sve vrste hrane

¥x [HRANA(x) => VOLI(SASA, x)]

2. Jabuke su hrana

HRANA(JABUKE)

3. Piletina je hrana

HRANA(PILETINA)

4. Hrana je sve što neko jede i ne otruje se

¥x ¥y [JEDE(y,x) ∧ ¬OTRUJE_SE(y,x) => HRANA(x)]

5. Srdjan jede kikiriki i još je živ

JEDE(SRDJAN, KIKIRIKI) ∧ ¬OTRUJE_SE(SRDJAN, KIKIRIKI)

6. Ceca jede sve što Srdjan jede

¥x [JEDE(SRDJAN,x) => JEDE(CECA,x)]

b) Pretvoriti iskaze u dobro formirane formule predikatske logike (u KNF)

- ¬HRANA(x) v VOLI(SASA, x)
- HRANA(JABUKA)
- HRANA(PILETINA)
- ¬JEDE(y, x1) v OTRUJE_SE(y,x1) v HRANA(x1)
- JEDE(SRDJAN, KIKIRIKI)
- ¬OTRUJE_SE(SRDJAN, KIKIRIKI)
- ¬JEDE(SRDJAN,x2) v JEDE(CECA,x2)

c) Rezolucijom pokazati da Saša voli kikiriki.

8. ¬VOLI(SASA, KIKIRIKI)

Iz 4. i 5.

4. ¬JEDE(y, x1) v OTRUJE_SE(y,x1) v HRANA(x1)

5. JEDE(SRDJAN, KIKIRIKI)

dobijamo

9. OTRUJE_SE(SRDJAN, KIKIRIKI) v HRANA(KIKIRIKI)

Iz 9. i 6.

6. ¬OTRUJE_SE(SRDJAN, KIKIRIKI)

9. OTRUJE_SE(SRDJAN, KIKIRIKI) v HRANA(KIKIRIKI)

dobijamo

10. HRANA (KIKIRIKI)

Iz 1. i 10.

1. ¬HRANA(x) v VOLI(SASA, x)

10. HRANA (KIKIRIKI)

dobijamo

11. VOLI(SASA, KIKIRIKI)

Protivrečnost !

Zadatak 7: Perica i Chop Suey (zaključivanje rezolucijom)

Ingredients

1 (1-pound) pork tenderloin
1/4 cup all-purpose flour
2 tablespoons vegetable oil, divided
2 cups thinly sliced bok choy
1 cup sliced celery
1 cup red bell pepper strips
1 cup sliced mushrooms
1 (8-ounce) can sliced water chestnuts, drained
2 garlic cloves, minced
1/4 cup fat-free, less-sodium chicken broth
1/4 cup low-sodium soy sauce
1 tablespoon cornstarch
1 tablespoon dry sherry
1/2 teaspoon ground ginger
2 cups hot cooked long-grain rice
1/4 cup sliced green onions

Poznate su činjenice

1. Perica voli sva laka jela
2. Jela francuske kuhinje su teška
3. Jela kineske kuhinje su laka
4. Chop Suey je jelo kineske kuhinje

Koristeći rezoluciju odgovoriti na pitanje: koje jelo voli Perica?

- Iskazi u predikatskoj logici

1. ¥x [Lako_jelo(x) => Perica_voli(x)]
2. ¥y [Francusko_jelo(y) => ¬Lako_jelo(y)]
3. ¥z [Kinesko_jelo(z) => Lako_jelo(z)]
4. Kinesko_jelo(CHOP_SUEY)

Dokazaćemo da postoji jelo koje Perica voli:

Эx [Perica_voli(x)]

- Dodajemo negaciju tvrdjenja:

$\neg \exists x$ [Perica_voli(x)]

- Prevodimo iskaze u konjuktivnu normalnu formu:

1. ¬Lako_jelo(x) v Perica_voli(x)
2. ¬Francusko_jelo(y) v ¬Lako_jelo(y)
3. ¬Kinesko_jelo(z) v Lako_jelo(z)
4. Kinesko_jelo(CHOP_SUEY)
5. ¬Perica_voli(w)

- Rezolucija uz strategiju *skupa podrške (eng. set-of-support)*
- Uvek se prvi biraju oni stavovi koji predstavljaju neagciju tvrdjenja ili stavove izvedene iz negacije tvrdjenja

Razmatramo redom:

1. ¬Lako_jelo(x) v Perica_voli(x)
2. ¬Francusko_jelo(y) v ¬Lako_jelo(y)
3. ¬Kinesko_jelo(z) v Lako_jelo(z)
4. Kinesko_jelo(CHOP_SUEY)
5. ¬Perica_voli(w)

Jedina moguća kombinacija 1., 5.

1., 5. $\xrightarrow{x = w}$ 6. ¬Lako_jelo(w)

Razmatramo stav 6. i ostale stavove

Razmatramo redom:

1. ¬Lako_jelo(x) v Perica_voli(x)
2. ¬Francusko_jelo(y) v ¬Lako_jelo(y)
3. ¬Kinesko_jelo(z) v Lako_jelo(z)
4. Kinesko_jelo(CHOP_SUEY)
5. ¬Perica_voli(w)
6. ¬Lako_jelo(w)

Jedina moguća kombinacija 3., 6.

3., 6. $\xrightarrow{x = w}$ 7. ¬Kinesko_jelo(w)

Razmatramo kombinacije sa stavom 7:

1. ¬Lako_jelo(x) v Perica_voli(x)
2. ¬Francusko_jelo(y) v ¬Lako_jelo(y)
3. ¬Kinesko_jelo(z) v Lako_jelo(z)
4. Kinesko_jelo(CHOP_SUEY)
5. ¬Perica_voli(w)
6. ¬Lako_jelo(w)
7. ¬Kinesko_jelo(w)

Jedina moguća kombinacija je 4., 7.

4., 7. $\xrightarrow{w\ =\ \text{CHOP\_SUEY}}$ NIL

Zadatak 8: Rezolucija uz izbor stavova po širini (zadatak 51)

b) Poznate su činjenice:

1. ¬P v R
2. ¬Q v R
3. P v Q

Pokazati da je R teorema, odnosno da R sledi iz prethodnih činjenica, koristeći rezoluciju uz strategiju izbora stavova *po širini*

- Strategija izbora po širini (eng. breadth-first)
- Razmatraju se redom sve moguće kombinacije postojećih stavova pre nego što se predje na novodobijene stavove

- Dodajemo negaciju teoreme

4. ¬R

1. ¬P v R

2. ¬Q v R

3. P v Q

4. ¬R

Stav 1. može se kombinovati sa stavovima 3. i 4. ali ne i sa stavom 2.:

1., 3. ⟶ 5. Q v R

1., 4. ⟶ 6. ¬P

1. ¬P v R
2. ¬Q v R
3. P v Q
4. ¬R
5. Q v R
6. ¬P

Stav 2. može se kombinovati sa stavovima 3. i 4.:

2., 3. → 7. P v R

2., 4. → 8. ¬Q

1. $\neg P \lor R$
2. $\neg Q \lor R$
3. $P \lor Q$
4. $\neg R$
5. $Q \lor R$
6. $\neg P$
7. $P \lor R$
8. $\neg Q$

Stavovi 3. i 4. se ne mogu kombinovati. Dalje je potrebno razmotriti kombinaciju stava 5. sa stavovima 1. do 4. ...

Teoreme i rezolucija

- Negirati teoremu koja se dokazuje i dodati je u listu aksioma
- Listu aksioma dovesti do KNF
- Dok se ne dobije prazna klauzula, ili dok se ne dobije par koji se ne može dalje razlagati, naći razlaganje klauzula, primeniti i dodati rezultat na listu klauzula
- Ako se dobije prazna klauzula, obavestiti da je teorema istinita u suprotnom obavestiti da je neistinita

Zadatak 10: Dokazivanje valjanosti formule (52)

Pokazati da je sledeća dobro formirana formula valjana primenom rezolucije:

$$(\forall x)\{P(x) \land [Q(A) \lor Q(B)]\}$$

$$\Rightarrow$$

$$(\exists x)\,[P(x) \land Q(x)\,]$$

- Dobro formirana formula naziva se valjanom ako je tačna za svaku moguću interpretaciju predikata
- Na stavove dobijene negiranjem polazne formule primenjuje se rezolucija, dok se ne dobije prazan stav

- Negiramo datu formulu:

$\neg \{(\text{¥}x)\{P(x) \wedge [Q(A) \vee Q(B)]\}$

=>

$(\exists x)\ [P(x) \wedge Q(x)\ ]\}$

- Prevodimo formulu u konjuktivnu normalnu formu kao pripremu za sprovodjenje rezolucije:

1. $P(x)$
2. $Q(A) \vee Q(B)$
3. $\neg P(y) \vee \neg Q(y)$

- Primenjujemo rezoluciju:

1., 3. $\xrightarrow{x\,=\,y}$ 4. $\neg Q(y)$

2., 4. $\xrightarrow{y\,=\,A}$ 5. $Q(B)$

4., 5. $\xrightarrow{y\,=\,B}$ NIL

Dodatno uraditi

- Zadatak 42
- Zadatak 47
- Zadatak 49
- Zadatak 51
- Zadatak 52 !

Hvala na pažnji!

Pismeni zadatak jednog beogradskog osnovca o ptici sovi.

Sova

Sova je ptica. Ona danju spava, a noću je potpuno slepa, kao slepi miš. Pošto ja ne znam puno o sovama,pisaću o životinji koju znam.
To je krava. Krava je sisar. Ona ima šest strana: gornju, donju, levu i desnu. Na zadnjoj strani je rep, na kraju repa je jedna četka. Ona služi da se krava brani od muva, da ne bi upadale u mleko. Mleko visi ispod krave. Na prednjoj strani je glava koja služi da na njoj rastu rogovi. Rogovi služe da bi krava bola. Glava služi i da bi na nju stala usta. Krava jede malo, jer ono što pojede, jede dva puta. Usta služe i da bi krava mukala. Ako ne muče, onda su joj usta puna.
Muška krava je vo. Vo nije sisar.
Meni još uvek nije jasno kako se dobija mleko.