



CG & GIS L@B
Design the future ...

Logika predikata I reda First Order Logic

Vladan Mihajlović
Aleksandar Milosavljević

Rezolucija

- Zasniva se na pravilu izvođenja poznato kao **princip rezolucije**:
- $\alpha \vee \beta$
- $\neg \beta \vee \gamma$
- -----
- $\alpha \vee \gamma$
- Način dokazivanja:
- Skupu aksioma i premisa se doda negacija tvrđenja koje treba dokazati, pa ako se dokaže kontradikcija, tvrđenje je tačno.
- Koristi se za dokazivanje teorema, tvrđenja i omogućuje davanje odgovora na pitanja.
- Princip rezolucije je logički zasnovan i kompletan sa jednim ograničenjem.
- Argumenti principa rezolucije su pojednostavljena verzija logike predikata, tzv. klauzule.



Elementi rezolucije

- Simboli, izrazi i atomi su isti kao u logici predikata, ali se koriste literali i klauzule umesto logičkih i kvantifikatorskih formula.
- Literal je atomska formula ili njena negacija.
- Klauzula je skup literala koji predstavlja njihovu disjunkciju.
- Klauzalni oblik je ekvivalentan disjunkciji u logici predikata.
- Klauzalni oblik (clausal form) je pojednostavljena verzija logike predikata.

Prevođenje u klauzalni oblik (1)

- 1. Eliminisanje implikacije
 - $\alpha \Rightarrow \beta \rightarrow \neg \alpha \vee \beta$
- 2. Sužavanje oblasti delovanja negacije
 - $\neg \neg \alpha \rightarrow \alpha$
 - $\neg(\alpha \wedge \beta) \rightarrow \neg \alpha \vee \neg \beta$
 - $\neg(\alpha \vee \beta) \rightarrow \neg \alpha \wedge \neg \beta$
 - $\neg \exists x \alpha \rightarrow \forall x \neg \alpha$
 - $\neg \forall x \alpha \rightarrow \exists x \neg \alpha$
- 3. Standardizacija promenljivih
 - Promenljive se preimenuju tako da svaki kvantifikator ima jedinstvenu promenljivu.
 - $(\forall x P(x,x)) \wedge (\exists x Q(x)) \rightarrow (\forall x P(x,x)) \wedge (\exists y Q(y))$



Prevođenje u klauzalni oblik (2)

- 4. Eliminisanje egzistencijalnih kvantifikatora (Skolemizacija)
 - Ako \exists nije u oblasti delovanja \forall , onda se uvodi Skolem konstanta
 - $\exists x P(x) \rightarrow P(A)$
 - A je nova konstanta
 - Ako \exists jeste u oblasti delovanja \forall , onda se uvodi Skolem funkcija:
 - $\forall y \exists x P(x, y) \rightarrow \forall y P(f(y), y)$
 - f je nova funkcija



Prevođenje u klauzalni oblik (3)

- 5. Izbacivanje univerzalnih kvantifikatora
 - $(\forall x P(x, y) \wedge Q(x)) \rightarrow P(x, y) \wedge Q(x)$
- 6. Prevođenje formule u konjunktivnu normalnu formu
 - $\alpha \vee (\beta \wedge \gamma) \rightarrow (\alpha \vee \beta) \wedge (\alpha \vee \gamma)$
- 7. Eliminisanje konjukcije
 - $\alpha \wedge (\beta \vee \gamma) \rightarrow \{ \alpha \} \text{ i } \{ \beta, \gamma \}$ (skup klauzula)
- 8. Preimenovanje promenljivih
 - Nijedna promenljiva se ne pojavljuje u vise od jedne klauzule.
- $\{P(x, y)\}, \{Q(x)\} \rightarrow \{P(x, y)\}, \{Q(z)\}$



Primer2 –

Prevođenje u klauzalni oblik

- Prevesti u klauzulni oblik formulu:
- $\forall x P(x) \Rightarrow ((\forall y P(y) \Rightarrow P(f(x, y)))$
 $\wedge \neg(\forall y Q(x, y) \Rightarrow P(y)))$
-



Primer2 –

Prevođenje u klauzalni oblik

- $\forall x P(x) \Rightarrow ((\forall y P(y) \Rightarrow P(f(x, y))) \wedge \neg(\forall y Q(x, y) \Rightarrow P(y)))$
- 1. $\forall x \neg P(x) \vee ((\forall y \neg P(y) \vee P(f(x, y))) \wedge \neg(\forall y \neg Q(x, y) \vee P(y)))$
- 2. $\forall x \neg P(x) \vee ((\forall y \neg P(y) \vee P(f(x, y))) \wedge (\exists y Q(x, y) \wedge \neg P(y)))$
- 3. $\forall x \neg P(x) \vee ((\forall y \neg P(y) \vee P(f(x, y))) \wedge (\exists z Q(x, z) \wedge \neg P(z)))$
- 4. $\forall x \neg P(x) \vee ((\forall y \neg P(y) \vee P(f(x, y))) \wedge (Q(x, g(x)) \wedge \neg P(g(x))))$
- 5. $\neg P(x) \vee ((\neg P(y) \vee P(f(x, y))) \wedge (Q(x, g(x)) \wedge \neg P(g(x))))$
- 6. $(\neg P(x) \vee \neg P(y) \vee P(f(x, y))) \wedge (\neg P(x) \vee Q(x, g(x)))$
 $\quad \quad \quad \wedge (\neg P(x) \vee \neg P(g(x)))$
- 7. $\{ \neg P(x), \neg P(y), P(f(x, y)) \}, \{ \neg P(x), Q(x, g(x)) \},$
 $\quad \quad \quad \{ \neg P(x), \neg P(g(x)) \}$
- 8. $\{ \neg P(x), \neg P(y), P(f(x, y)) \}, \{ \neg P(z), Q(z, g(z)) \},$
 $\quad \quad \quad \{ \neg P(v), \neg P(g(v)) \}$
-



Izvođenje I dokazivanje rezolucijom

- Klausula α koja se nalazi u skupu klauzula koje se dobijaju procesom rezolucije je:
 - a) α element polaznog skupa klauzula
 - b) α je dobijen primenom principa rezolucije na klauzule koje postoje u skupu klauzula
- Skupu klauzula se doda negacija tvrđenje (prevedena u klauzalni oblik), i primenjuje se pravilo rezolucije.
- Ako se dobije prazna klauzula, onda tvrđenje logički sleduje iz polaznog skupa klauzula.
- Prazna klauzula pokazuje da je skup iskaza sastavljen od polaznog skupa klauzula i negacije tvrđenja kontradiktoran.



Primer rezolucije – Problem

- 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. Srđan jede kikiriki i još je živ.
- 6. Ceca jede sve što Srđan jede.
- Pokazati rezolucijom da:
- Saša voli kikiriki i piletinu.



Primer rezolucije – Izdvajanje konstanti

- 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. **Srđan** jede **kikiriki** i još je živ.
- 6. **Ceca** jede sve što **Srđan** jede.
- **Saša** voli **kikiriki** i **piletinu**.

- Konstante su:

- | | |
|-------------------|---------------|
| ● Saša | Jabuka |
| ● Piletina | Srđan |
| ● Kikiriki | Ceca |



Primer rezolucije – Izdvajanje predikata

- 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. Srđan **jede** kikiriki i još je **živ**.
 - 6. Ceca **jede** sve što Srđan **jede**.
 - Saša **voli** kikiriki i piletinu.
-
- Predikati su:
 - **voli**
 - **jede**
 - **živ**
- | |
|------------------|
| hrana |
| otruje_se |



Primer rezolucije – Uprošćavanje problema

- 4. Hrana je sve ono što neko jede i ne **otruje se**.
- 5. Srđan jede kikiriki i još je **živ**.
- Da bi se uspostavila veza između predikata otruje se i živ, potrebno je dodati novi iskaz:
- Ko je **živ** nije **se otrovao**.
- Međutim, ovim se povećava složenost baze znanja, pa je bolje umesto podrazumevanog iskaza predikat **živ** zameniti tvrđenjem **nije se otrovao**.
- 4. Hrana je sve ono što neko jede i ne **otruje se**.
- 5. Srđan jede kikiriki i nije **se otrovao**.



Primer rezolucije – Prerađeni skup iskaza

- 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. Srđan jede kikiriki i nije **se otrovao**.
 - 6. Ceca **jede** sve što Srđan **jede**.
 - Saša **voli** kikiriki i piletinu.
-
- Predikati su:
 - **voli** **hrana**
 - **jede** **otruje_se**



Primer rezolucije – Format predikata

- Predikat ***voli*** uspostavlja vezu između osobe ***x*** i hrane ***y*** koju voli osoba ***x***:
- ***Voli(x, y)***
- Funkcija hrana označava da se neki objekat realnog sveta može klasifikovati kao hrana.
- ***Hrana(x)***
- Predikat ***otruje se*** povezuje osobu ***x*** koja se otrovala hranom ***y***:
- ***Otruje_se(x, y)***
- Predikat ***jede*** iskazuje da osoba ***x*** jede određenu vrstu hrane ***y***:
- ***Jede(x, y)***



Primer rezolucije – Prevođenje u logiku predikata

- 1. Saša voli sve vrste hrane.
- $\forall x \text{ *Hrana*(x)} \Rightarrow \text{*Voli*(Saša, x)}$
- 2. Jabuke su hrana.
- *Hrana*(Jabuka)
- 3. Piletina je hrana.
- *Hrana*(Piletina)
- 4. Hrana je sve ono što neko jede i ne otruje se.
- $\forall x \forall y \text{ *Jede*(x, y) } \wedge \neg \text{*Otruje_se*(x, y)} \Rightarrow \text{*Hrana*(y)}$
- 5. Srđan jede kikiriki i nije se otrovao.
- $\text{*Jede*(Srđan, Kikiriki)} \wedge \neg \text{*Otruje_se*(Srđan, Kikiriki)}$
- 6. Ceca jede sve što Srđan jede.
- $\forall x \text{ *Jede*(Srđan, x)} \Rightarrow \text{*Jede*(Ceca, x)}$
- Saša voli kikiriki i piletinu.
- $\text{*Voli*(Saša, Kikiriki)} \wedge \text{*Voli*(Saša, Piletina)}$



Primer rezolucije – Skup polaznih formula

- 1. $\forall x \text{ Hrana}(x) \Rightarrow \text{Voli}(\text{Saša}, x)$
- 2. $\text{Hrana}(\text{Jabuka})$
- 3. $\text{Hrana}(\text{Piletina})$
- 4. $\forall x \forall y \text{ Jede}(x, y) \wedge \neg \text{Otruje_se}(x, y) \Rightarrow \text{Hrana}(y)$
- 5. $\text{Jede}(\text{Srđan}, \text{Kikiriki})$
- $\wedge \neg \text{Otruje_se}(\text{Srđan}, \text{Kikiriki})$
- 6. $\forall x \text{ Jede}(\text{Srđan}, x) \Rightarrow \text{Jede}(\text{Ceca}, x)$
- 7. $\neg(\text{Voli}(\text{Saša}, \text{Kikiriki}) \wedge \text{Voli}(\text{Saša}, \text{Piletina}))$



Primer rezolucije – Eliminisanje implikacije

- 1. $\forall x \text{ *Hrana*(x)} \Rightarrow \text{Voli}(\text{Saša}, x)$
- $\rightarrow \forall x \neg \text{Hrana}(x) \vee \text{Voli}(\text{Saša}, x)$
- 4. $\forall x \forall y \text{ Jede}(x, y) \wedge \neg \text{Otruje_se}(x, y)$
 $\Rightarrow \text{Hrana}(y)$
- $\rightarrow \forall x \forall y \neg(\text{Jede}(x, y)$
 $\wedge \neg \text{Otruje_se}(x, y)) \vee \text{Hrana}(\text{Piletina})$
- 6. $\forall x \text{ Jede}(\text{Srđan}, x) \Rightarrow \text{Jede}(\text{Ceca}, x)$
- $\rightarrow \forall x \neg \text{Jede}(\text{Srđan}, x) \vee \text{Jede}(\text{Ceca}, x)$



Primer rezolucije – Prevođenje u klauzulni oblik (korak 1)

- 1. $\forall x \neg Hrana(x) \vee Voli(Saša, x)$
- 2. $Hrana(Jabuka)$
- 3. $Hrana(Piletina)$
- 4. $\forall x \forall y \neg (Jede(x, y) \wedge \neg Otruje_se(x, y))$
● $\vee Hrana(y)$
- 5. $Jede(Srđan, Kikiriki)$
● $\wedge \neg Otruje_se(Srđan, Kikiriki)$
- 6. $\forall x \neg Jede(Srđan, x) \vee Jede(Ceca, x)$
- 7. $\neg (Voli(Saša, Kikiriki)$
● $\wedge Voli(Saša, Piletina))$



Primer rezolucije – Sužavanje oblasti delovanja negacije

- 4. $\forall x \forall y \neg(Jede(x, y) \wedge$
- $\neg Otruje_se(x, y)) \vee Hrana(Piletina)$
- $\rightarrow \forall x \forall y \neg Jede(x, y) \vee$
- $\neg \neg Otruje_se(x, y)) \vee Hrana(Piletina)$
- $\rightarrow \forall x \forall y \neg Jede(x, y) \vee Otruje_se(x, y)$
- $\vee Hrana(y)$
- $\Gamma. \neg(Voli(Saša, Kikiriki)$
- $\wedge Voli(Saša, Piletina))$
- $\rightarrow \neg Voli(Saša, Kikiriki)$
- $\vee \neg Voli(Saša, Piletina)$



Primer rezolucije – Prevođenje u klauzulni oblik (korak 2)

- 1. $\forall x \neg Hrana(x) \vee Voli(Saša, x)$
- 2. $Hrana(Jabuka)$
- 3. $Hrana(Piletina)$
- 4. $\forall x \forall y \neg Jede(x, y) \vee Otruje_se(x, y)$
● $\vee Hrana(y)$
- 5. $Jede(Srđan, Kikiriki)$
● $\wedge \neg Otruje_se(Srđan, Kikiriki)$
- 6. $\forall x \neg Jede(Srđan, x) \vee Jede(Ceca, x)$
- 7. $\neg Voli(Saša, Kikiriki)$
● $\vee \neg Voli(Saša, Piletina)$



Primer rezolucije – Prevođenje u klauzulni oblik (korak 5)

- 1. $\neg \text{Hrana}(x) \vee \text{Voli}(\text{Saša}, x)$
- 2. $\text{Hrana}(\text{Jabuka})$
- 3. $\text{Hrana}(\text{Piletina})$
- 4. $\neg \text{Jede}(x, y) \vee \text{Otruje_se}(x, y) \vee \text{Hrana}(y)$
- 5. $\text{Jede}(\text{Srđan}, \text{Kikiriki})$
- $\wedge \neg \text{Otruje_se}(\text{Srđan}, \text{Kikiriki})$
- 6. $\neg \text{Jede}(\text{Srđan}, x) \vee \text{Jede}(\text{Ceca}, x)$
- $\Gamma. \neg \text{Voli}(\text{Saša}, \text{Kikiriki})$
- $\vee \neg \text{Voli}(\text{Saša}, \text{Piletina})$



Primer rezolucije – Prevođenje u klauzulni oblik (korak 7)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(x, y), Otruje_se(x, y), Hrana(y)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 6. $\{\neg Jede(Srđan, x), Jede(Ceca, x)\}$
- $\Gamma. \{\neg Voli(Saša, Kikiriki),$
 $\neg Voli(Saša, Piletina)\}$



Primer rezolucije – Prevođenje u klauzulni oblik (korak 8)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(y, z), Otruje_se(y, z), Hrana(z)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 6. $\{\neg Jede(Srđan, v), Jede(Ceca, v)\}$
- $\Gamma. \{\neg Voli(Saša, Kikiriki),$
 $\neg Voli(Saša, Piletina)\}$



Primer rezolucije – Zaključivanje (1)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(y, z), Otruje_se(y, z), Hrana(z)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- 6. $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 7. $\{\neg Jede(Srđan, v), Jede(Ceca, v)\}$
- 8. $\{\neg Voli(Saša, Kikiriki), \neg Voli(Saša, Piletina)\}$
- Iz 4. i 5. unifikacijom promenljivih $\{y/Srđan, z/Kikiriki\}$ dobija se:
- 9. $\{Otruje_se(Srđan, Kikiriki), Hrana(Kikiriki)\}$



Primer rezolucije – Zaključivanje (2)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(y, z), Otruje_se(y, z), Hrana(z)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- 6. $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 7. $\{\neg Jede(Srđan, v), Jede(Ceca, v)\}$
- 8. $\{\neg Voli(Saša, Kikiriki), \neg Voli(Saša, Piletina)\}$
- 9. $\{Otruje_se(Srđan, Kikiriki), Hrana(Kikiriki)\}$
- Iz 9. i 6. dobija se:
- 10. $\{Hrana(Kikiriki)\}$



Primer rezolucije – Zaključivanje (3)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(y, z), Otruje_se(y, z), Hrana(z)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- 6. $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 7. $\{\neg Jede(Srđan, v), Jede(Ceca, v)\}$
- 8. $\{\neg Voli(Saša, Kikiriki), \neg Voli(Saša, Piletina)\}$
- 9. $\{Otruje_se(Srđan, Kikiriki), Hrana(Kikiriki)\}$
- 10. $\{Hrana(Kikiriki)\}$
- Iz 1. i 10. unifikacijom promenljivih $\{x/Kikiriki\}$ dobija se:
- 11. $\{Voli(Saša, Kikiriki)\}$



Primer rezolucije – Zaključivanje (4)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(y, z), Otruje_se(y, z), Hrana(z)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- 6. $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 7. $\{\neg Jede(Srđan, v), Jede(Ceca, v)\}$
- 8. **$\{\neg Voli(Saša, Kikiriki), \neg Voli(Saša, Piletina)\}$**
- 9. $\{Otruje_se(Srđan, Kikiriki), Hrana(Kikiriki)\}$
- 10. $\{Hrana(Kikiriki)\}$
- 11. **$\{Vol i(Saša, Kikiriki)\}$**
- Iz 8. i 11. dobija se:
- 12. **$\{\neg Vol i(Saša, Piletina)\}$**



Primer rezolucije – Zaključivanje (5)

- 1. $\{\neg Hrana(x), Voli(Saša, x)\}$
- 2. $\{Hrana(Jabuka)\}$
- 3. $\{Hrana(Piletina)\}$
- 4. $\{\neg Jede(y, z), Otruje_se(y, z), Hrana(z)\}$
- 5. $\{Jede(Srđan, Kikiriki)\}$
- 6. $\{\neg Otruje_se(Srđan, Kikiriki)\}$
- 7. $\{\neg Jede(Srđan, v), Jede(Ceca, v)\}$
- 8. $\{\neg Voli(Saša, Kikiriki), \neg Voli(Saša, Piletina)\}$
- 9. $\{Otruje_se(Srđan, Kikiriki), Hrana(Kikiriki)\}$
- 10. $\{Hrana(Kikiriki)\}$
- 11. $\{Voli(Saša, Kikiriki)\}$
- 12. $\{\neg Voli(Saša, Piletina)\}$
- Iz 1. i 3. unifikacijom promenljivih $\{x/Piletina\}$ dobija se:
- 13. $\{Voli(Saša, Piletina)\}$



Primer rezolucije – Zaključivanje (6)

- 1. $\{\neg \text{Hrana}(x), \text{Voli}(\text{Saša}, x)\}$
- 2. $\{\text{Hrana}(\text{Jabuka})\}$
- 3. $\{\text{Hrana}(\text{Piletina})\}$
- 4. $\{\neg \text{Jede}(y, z), \text{Otruje_se}(y, z), \text{Hrana}(z)\}$
- 5. $\{\text{Jede}(\text{Srđan}, \text{Kikiriki})\}$
- 6. $\{\neg \text{Otruje_se}(\text{Srđan}, \text{Kikiriki})\}$
- 7. $\{\neg \text{Jede}(\text{Srđan}, v), \text{Jede}(\text{Ceca}, v)\}$
- 8. $\{\neg \text{Voli}(\text{Saša}, \text{Kikiriki}), \neg \text{Voli}(\text{Saša}, \text{Piletina})\}$
- 9. $\{\text{Otruje_se}(\text{Srđan}, \text{Kikiriki}), \text{Hrana}(\text{Kikiriki})\}$
- 10. $\{\text{Hrana}(\text{Kikiriki})\}$
- 11. $\{\text{Voli}(\text{Saša}, \text{Kikiriki})\}$
- 12. $\{\neg \text{Voli}(\text{Saša}, \text{Piletina})\}$
- 13. $\{\text{Voli}(\text{Saša}, \text{Piletina})\}$
- Iz 12. i 13. dobija se:
- 14. $\{\}$



Jezik logike predikata

- Jezik logike predikata je formalni jezik, sastavljen od rečenica (formula).
- Logika predikata je proširenje logike iskaza (propositional logic).
- Logika predikata prvog reda ne dozvoljava kvantifikaciju predikata i funkcija.
- Istinitost formula predikatske logike se utvrđuje pomoću uzastopne primene relacija iz istinitonosnih tablica, počev od najdublje ugnježdene formule (od unutrašnjosti prema spoljašnjosti formule).
- Azbuka logike predikata:
 - Specijalni znaci
 - Simboli
 - Izrazi
 - Rečenice (formule)



Algoritam unifikacije

- P i Q su formule, a S je lista smena.
- Ako su P i Q jednaki, izlaz S .
- Ako su P i Q različiti atomi, izlaz nil .
- Ako je jedan od uzoraka promenljiva, pozvati funkciju *unifikacije promenljivih*.
- U suprotnom, P i Q moraju biti liste.
 - Ako su liste različite dužine, izlaz nil .
 - Zovi funkciju *unifikacije promenljivih* za odgovarajuće 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 obrađene, izlaz S , a ako nisu preći na sledeću komponentu.



Algoritam unifikacije promenljivih

- **var** je promenljiva, a **pat** šablon kojim se ona zamenjuje (konstanta, funkcija, predikat)
- Ako **var** ima dodeljenu vrednost u S, vrati rezultat unifikovanja te vrednosti sa **pat**.
- Ako je **var=pat**, izlaz S.
- U ostalim slučajevima, proveriti da li se **var** javlja u **pat**.
 - Ako da, izlaz *nil*.
 - Ako ne, dodaj novu smenu **var/pat** listi S i vrati S.



Pravila zaključivanja

- MODUS PONENS (MP)
- MODUS TOLENS (MT)
- Eliminacija konjunkcije – AND ELIMINATION (AE)
- Uvođenje konjunkcije – AND INTRODUCTION (AI)
- Eliminisanje univerzalnog kvantifikatora – UNIVERSAL INSTANTIATION (UI)
- Eliminisanje egzistencijalnog kvantifikatora – EXISTENTIAL INSTANTIATION (EI)
- Zaključak **P** je izvodljiv iz skupa premisa ako i samo ako
 - **P** je član Δ
 - **P** je rezultat primene pravila izvođenja na formule iz Δ



MODUS PONENS (MP)

- **MODUS PONENS (MP)**
- $P \Rightarrow Q \leftarrow$ premisa
- $P \leftarrow$ premisa
- -----
- $Q \leftarrow$ zaključak
-
- $\text{Nosi}(x, y) \Rightarrow \text{Iznad}(y, x)$
- $\text{Nosi}(A, B)$
- $\text{Iznad}(B, A)$



MODUS TOLENS (MT)

- **MODUS TOLENS (MT)**
- $P \Rightarrow Q \leftarrow$ premisa
- $\neg Q \leftarrow$ premisa
- -----
- $\neg P \leftarrow$ zaključak
-
- $\text{Nosi}(x, y) \Rightarrow \text{Iznad}(y, x)$
- $\neg \text{Iznad}(B, A)$
- $\neg \text{Nosi}(A, B)$



Eliminacija konjunkcije

- **AND ELIMINATION (AE)**
- Eliminacija konjunkcije
- $P \wedge Q \leftarrow$ premisa
- -----
- $P \leftarrow$ zaključak
- $Q \leftarrow$ zaključak
-
- $\text{Nosi}(A, B) \wedge \text{Nosi}(B, C)$
- $\text{Nosi}(A, B)$
- $\text{Nosi}(B, C)$



Uvođenje konjunkcije

- **AND INTRODUCTION (AI)**

- Uvođenje konjunkcije

- P \leftarrow premisa

- Q \leftarrow premisa

- -----

- $P \wedge Q$ \leftarrow zaključak

-

- $\text{Nosi}(A, B)$

- $\text{Nosi}(B, C)$

- $\text{Nosi}(A, B) \wedge \text{Nosi}(B, C)$



Eliminisanje univerzalnog kvantifikatora

- **UNIVERSAL INSTANTIATION (UI)**
- Eliminisanje univerzalnog kvantifikatora
- $\forall x_i P(x_1, \dots, x_i, \dots, x_n) \leftarrow$ premisa
- -----
- $P(x_1, \dots, t, \dots, x_n) \leftarrow$ zaključak
- t je izraz u kome se x_i ne javlja kao slobodna promenljiva (konstanta)
-
- $\forall x \text{ Nosi}(x, B)$
- $\text{Nosi}(A, B)$
- $\forall z \text{ Iznad}(y, z)$
- $\text{Iznad}(y, C)$



Eliminisanje egzistencijalnog kvantifikatora

- **EXISTENTIAL INSTANTIATION (EI)**
- Eliminisanje egzistencijalnog kvantifikatora
- $\exists x_i P(x_1, \dots, x_i, \dots, x_n)$ \leftarrow premisa
- -----
- $P(x_1, \dots, f(y_1, \dots, y_n), \dots, x_n)$ \leftarrow zaključak
- $f(y_1, \dots, y_n)$ je funkcija, a y_1, \dots, y_n su slobodne promenljive u formuli $P(x_1, \dots, x_i, \dots, x_n)$

- $\exists x \text{ Nosi}(x, B)$
- $\text{Nosi}(A, B)$
- $\exists z \text{ Iznad}(y, z)$
- $\text{Iznad}(y, f(y))$



Primer 3 – Programer

- Dato je pravilo:
- Svestran programer je onaj ko zna da programira na jeziku niskog nivoa, višeg nivoa i funkcionalnom jeziku.
- Dati su iskazi:
- Zoran je isprogramirao PIC kontroler. Zoran je napravio projekat “B stablo” na jeziku C++.
- Zoran je napravio igricu “Trka” koristeći Lisp.
- Dokazati da je Zoran svestran programer.



Primer 3 – Identifikacija konstanti

- Svestran programer je onaj ko zna da programira na jeziku niskog nivoa, višeg nivoa i funkcionalnom jeziku.
- Zoran je isprogramirao PIC kontroler.
- Zoran je napravio projekat “B stablo” na jeziku C++.
- Zoran je napravio igricu “Trka” koristeći Lisp.
- Konstante:
 - PIC
 - B stablo
 - Jezik C++
 - Trka
 - Lisp
 - Zoran



Primer 3 – Identifikacija predikata

- Svestran programer je onaj ko zna da programira na jeziku niskog nivoa, višeg nivoa i funkcionalnom jeziku.
- Zoran je isprogramirao PIC kontroler.
- Zoran je napravio projekat “B stablo” na jeziku C++.
- Zoran je napravio igricu “Trka” koristeći Lisp.
- Predikati:
 - Svestran programer
 - Jezik niskog nivoa
 - Jezik višeg nivoa
 - Funkcionalni jezik
 - Programira
 - Isprogramirao
 - Napravi



Primer 3 – Format predikata

- Svestran programer je atribut koji opisuje čoveka
 - SvestranProgramer(x)
- Jezik nižeg nivoa je atribut programskog jezika
 - JezikNižegNivoa(x)
- Jezik višeg nivoa je atribut programskog jezika
 - JezikVišegNivoa(x)
- Funkcionalni jezik je atribut programskog jezika
 - FunkcionalniJezik(x)
- x programira na jeziku y
 - Programira(x, y)
- x je isprogramirao hardversku komponentu y
 - Isprogramirao(x, y)
- Napravi x softver y koristeći jezik z.
 - Napravi(x, y, z)



Primer 3 – Formiranje formula

- Svestran programer je onaj ko zna da programira na jeziku niskog nivoa, višeg nivoa i funkcionalnom jeziku.
- $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
- Zoran je isprogramirao PIC kontroler.
- **Isprogramirao(Zoran, PIC) ... Δ**
- Zoran je napravio projekat “B stablo” na jeziku C++.
- **Napravio(Zoran, BStablo, JezikC++) ... Δ**
- Zoran je napravio igricu “Trka” koristeći Lisp.
- **Napravio(Zoran, Trka, Lisp) ... Δ**



Primer 3 – Proširenje baze znanja

- PIC kontroler koristi Asembler.
- **Koristi(PIC, Asembler) ... Δ**
- Asembler je jezik niskog nivoa.
- **JezikNiskogNivoa(Asembler) ... Δ**
- Jezik C++ je jezik višeg nivoa.
- **JezikVisokogNivoa(jezikC++) ... Δ**
- Lisp je funkcionalni jezik.
- **FunkcionalniJezik(Lisp) ... Δ**
- Ako neko isprogramira hardversku komponentu koja koristi neki programski jezik onda on programira u tom jeziku.
- **$\forall x \forall y \forall z \text{ Isprogramira}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x, z) \dots \Delta$**
- Ako neko napravio nešto koristeći neki programski jezik onda on programira u tom jeziku.
- **$\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x, z) \dots \Delta$**



Primer 3 – Zaključivanje (1)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$



Primer 3 – Zaključivanje (2)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
 - Primenom UI na pravilo 9 uz smene $\{x/\text{Zoran}, y/\text{PIC}, z/\text{Assembler}\}$ dobijamo:
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler}) \dots \Delta$



Primer 3 – Zaključivanje (3)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoaa}(x) \wedge \text{JezikVišegNivoaa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Asembler}) \dots \Delta$
6. $\text{JezikNiskogNivoaa}(\text{Asembler}) \dots \Delta$
7. $\text{JezikVisokogNivoaa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Asembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Asembler})$
- **Primenom AI na činjenice 2 i 5 dobijamo:**
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Asembler})$



Primer 3 – Zaključivanje (4)

1. $\forall x \forall y \forall z \forall u \text{JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
 - Primenom MP na činjenice 11 i 12 dobijamo:
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$



Primer 3 – Zaključivanje (5)

1. $\forall x \forall y \forall z \forall u \text{JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$
- **Primenom UI na pravilo 10 uz smene $\{x/\text{Zoran}, y/\text{BStablo}, z/\text{JezikC++}\}$ dobijamo:**
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$



Primer 3 – Zaključivanje (6)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$
- **Primenom MP na činjenice 14 i 3 dobijamo:**
15. $\text{Programira}(\text{Zoran}, \text{JezikC++})$

Primer 3 – Zaključivanje (7)

1. $\forall x \forall y \forall z \forall u \text{JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$
15. $\text{Programira}(\text{Zoran}, \text{JezikC++})$
- $\text{Primenom UI na pravilo 10 uz smene } \{x/\text{Zoran}, y/\text{Trka}, z/\text{Lisp}\} \text{ dobijamo:}$
16. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Lisp})$



Primer 3 – Zaključivanje (8)

1. $\forall x \forall y \forall z \forall u \text{JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$
15. $\text{Programira}(\text{Zoran}, \text{JezikC++})$
16. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Lisp})$
 - Primenom MP na činjenice 16 i 4 dobijamo:
17. $\text{Programira}(\text{Zoran}, \text{Lisp})$



Primer 3 – Zaključivanje (9)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoaa}(x) \wedge \text{JezikVišegNivoaa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoaa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoaa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$
15. $\text{Programira}(\text{Zoran}, \text{JezikC++})$
16. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Lisp})$
17. $\text{Programira}(\text{Zoran}, \text{Lisp})$
- **Primenom UI na pravilo 1 uz smene $\{x/\text{Assembler}, y/\text{Jezik C++}, z/\text{Lisp}, u/\text{Zoran}\}$ dobijamo:**
18. $\text{JezikNižegNivoaa}(\text{Assembler}) \wedge \text{JezikVišegNivoaa}(\text{JezikC++}) \wedge \text{FunkcionalniJezik}(\text{Lisp}) \wedge \text{Programira}(\text{Zoran}, \text{Assembler}) \wedge \text{Programira}(\text{Zoran}, \text{JezikC++}) \wedge \text{Programira}(\text{Zoran}, \text{Lisp}) \Rightarrow \text{SvestranProgramer}(\text{Zoran})$



Primer 3 – Zaključivanje (10)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoa}(x) \wedge \text{JezikVišegNivoa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Assembler}) \dots \Delta$
6. $\text{JezikNiskogNivoa}(\text{Assembler}) \dots \Delta$
7. $\text{JezikVisokogNivoa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Assembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Assembler})$
13. $\text{Programira}(\text{Zoran}, \text{Assembler})$
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$
15. $\text{Programira}(\text{Zoran}, \text{JezikC++})$
16. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Lisp})$
17. $\text{Programira}(\text{Zoran}, \text{Lisp})$
18. $\text{JezikNižegNivoa}(\text{Assembler}) \wedge \text{JezikVišegNivoa}(\text{JezikC++}) \wedge \text{FunkcionalniJezik}(\text{Lisp}) \wedge \text{Programira}(\text{Zoran}, \text{Assembler}) \wedge \text{Programira}(\text{Zoran}, \text{JezikC++}) \wedge \text{Programira}(\text{Zoran}, \text{Lisp}) \Rightarrow \text{SvestranProgramer}(\text{Zoran})$
- **Primenom AI na činjenice 6, 7, 8, 13, 15, 17 dobijamo:**
19. $\text{JezikNižegNivoa}(\text{Assembler}) \wedge \text{JezikVišegNivoa}(\text{JezikC++}) \wedge \text{FunkcionalniJezik}(\text{Lisp}) \wedge \text{Programira}(\text{Zoran}, \text{Assembler}) \wedge \text{Programira}(\text{Zoran}, \text{JezikC++}) \wedge \text{Programira}(\text{Zoran}, \text{Lisp})$



Primer 3 – Zaključivanje (11)

1. $\forall x \forall y \forall z \forall u \text{ JezikNižegNivoaa}(x) \wedge \text{JezikVišegNivoaa}(y) \wedge \text{FunkcionalniJezik}(z) \wedge \text{Programira}(u,x) \wedge \text{Programira}(u,y) \wedge \text{Programira}(u,z) \Rightarrow \text{SvestranProgramer}(u) \dots \Delta$
2. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \dots \Delta$
3. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \dots \Delta$
4. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \dots \Delta$
5. $\text{Koristi}(\text{PIC}, \text{Asembler}) \dots \Delta$
6. $\text{JezikNiskogNivoaa}(\text{Asembler}) \dots \Delta$
7. $\text{JezikVisokogNivoaa}(\text{JezikC++}) \dots \Delta$
8. $\text{FunkcionalniJezik}(\text{Lisp}) \dots \Delta$
9. $\forall x \forall y \forall z \text{ Isprogramirao}(x, y) \wedge \text{Koristi}(y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
10. $\forall x \forall y \forall z \text{ Napravio}(x, y, z) \Rightarrow \text{Programira}(x,z) \dots \Delta$
11. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Asembler}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Asembler})$
12. $\text{Isprogramirao}(\text{Zoran}, \text{PIC}) \wedge \text{Koristi}(\text{PIC}, \text{Asembler})$
13. $\text{Programira}(\text{Zoran}, \text{Asembler})$
14. $\text{Napravio}(\text{Zoran}, \text{BStablo}, \text{JezikC++}) \Rightarrow \text{Programira}(\text{Zoran}, \text{JezikC++})$
15. $\text{Programira}(\text{Zoran}, \text{JezikC++})$
16. $\text{Napravio}(\text{Zoran}, \text{Trka}, \text{Lisp}) \Rightarrow \text{Programira}(\text{Zoran}, \text{Lisp})$
17. $\text{Programira}(\text{Zoran}, \text{Lisp})$
18. $\text{JezikNižegNivoaa}(\text{Asembler}) \wedge \text{JezikVišegNivoaa}(\text{JezikC++}) \wedge \text{FunkcionalniJezik}(\text{Lisp}) \wedge \text{Programira}(\text{Zoran}, \text{Asembler}) \wedge \text{Programira}(\text{Zoran}, \text{JezikC++}) \wedge \text{Programira}(\text{Zoran}, \text{Lisp}) \Rightarrow \text{SvestranProgramer}(\text{Zoran})$
19. $\text{JezikNižegNivoaa}(\text{Asembler}) \wedge \text{JezikVišegNivoaa}(\text{JezikC++}) \wedge \text{FunkcionalniJezik}(\text{Lisp}) \wedge \text{Programira}(\text{Zoran}, \text{Asembler}) \wedge \text{Programira}(\text{Zoran}, \text{JezikC++}) \wedge \text{Programira}(\text{Zoran}, \text{Lisp})$
 - Primenom MP na činjenice 18 i 19 dobijamo:
20. $\text{SvestranProgramer}(\text{Zoran})$



Primer 4 – Parovi

- Dati sledeći iskazi:
- Devojke vole momke koji ne piju, koji su duhoviti i pametni. Momci vole devojke koje su zgodne i inteligentne. Milan vodi zdrav život i pametan je. Ljudi smatraju da je Milan duhovit. Milena je član Mense. Ljudi smatraju da je Milena zgodna. Ako momak voli devojku i ona njega onda su oni dobar par.
- Dokazati da su Milan i Milena dobar par.



Primer4 – Identifikacija konstanti

- Devojke vole momke koji ne piju, koji su duhoviti i pametni.
- Momci vole devojke koje su zgodne i inteligentne.
- Milan vodi zdrav život i pametan je.
- Ljudi smatraju da je Milan duhovit.
- Milena je član Mense.
- Ljudi smatraju da je Milena zgodna.
- Ako momak voli devojku i ona njega onda su oni dobar par.
- Dokazati da su Milan i Milena dobar par.
- Konstante
- Milan
- Milena



Primer4 – Identifikacija predikata

- Devojke vole momke koji ne piju, koji su duhoviti i pametan.
 - Momci vole devojke koje su zgodne i inteligentne.
 - Milan vodi zdrav život i pametan je.
 - Ljudi smatraju da je Milan duhovit.
 - Milena je član Mense.
 - Ljudi smatraju da je Milena zgodna.
 - Ako momak voli devojku i ona njega onda su oni dobar par.
 - Dokazati da su Milan i Milena dobar par.
 - Predikati
 - Devojka
 - Pije
 - Pametan
 - Inteligentan
 - ČlanMense
 - DuhovitZa
 - Voli
- | |
|----------------|
| Momak |
| Duhovit |
| Zgodan |
| VodiZdravŽivot |
| Smatra |
| ZgodanZa |
| DobarPar |



Primer4 – Format predikata

- Devojka i Momak su atributi koji opisuju pol čoveka
 - Devojka(x) Momak(x)
- Pije se odnosi na osobina čoveka da voli da popije
 - Pije(x)
- Duhovit, Pametan, Zgodan i Inteligentan su atributi čoveka
 - Duhovit(x) Pametan(x) Zgodan(x) Inteligentan(x)
- Vodi zdrav život je način koji opisuje ponašanje čoveka
 - VodiZdravŽivot(x)
- Član Mense označava pripadnost jedne osobe organizaciji Mensa
 - ČlanMense(x)
- Smatra je odnos da jedna osoba misli nešto o drugoj osobi
 - Smatra(x, y)
- DuhovitZa i ZgodanZa su odnosi koji predstavljaju stavove jedne osobe o drugoj
 - DuhovitZa(x, y) ZgodanZa(x, y)
- Voli je osećanje jedne osobe prema drugoj
 - Voli(x, y)
- Dobar par opisuje odnos između dve osobe
 - DobarPar(x, y)



Primer4 – Formiranje formula

- Devojke vole momke koji ne piju, koji su duhoviti i pametan.
- $\forall x \forall y \text{ Devojka}(x) \wedge \text{Momak}(y) \wedge \neg \text{Pije}(y) \wedge \text{Duhovit}(x) \wedge \text{Pametan}(y) \Rightarrow \text{Voli}(x, y) \dots \Delta$
- Momci vole devojke koje su zgodne i inteligentne.
- $\forall x \forall y \text{ Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Zgodan}(y) \wedge \text{Inteligentan}(y) \Rightarrow \text{Voli}(x, y) \dots \Delta$
- Milan vodi zdrav život i pametan je.
- $\text{VodiZdravŽivot}(\text{Milan}) \wedge \text{Pametan}(\text{Milan}) \dots \Delta$
- Ljudi smatraju da je Milan duhovit.
- $\forall x \text{ Smatra}(x, \text{Milan}) \Rightarrow \text{DuhovitZa}(\text{Milan}, x) \dots \Delta$
- Milena je član Mense.
- $\text{ČlanMense}(\text{Milena}) \dots \Delta$
- Ljudi smatraju da je Milena zgodna.
- $\forall x \text{ Smatra}(x, \text{Milena}) \Rightarrow \text{ZgodanZa}(\text{Milena}, x) \dots \Delta$
- Ako momak voli devojku i ona njega onda su oni dobar par.
- $\forall x \forall y \text{ Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Voli}(x, y) \wedge \text{Voli}(y, x) \Rightarrow \text{DobarPar}(x, y) \dots \Delta$
- Milan i Milena su dobar par.
- $\text{DobarPar}(\text{Milan}, \text{Milena}) \dots \Gamma$



Primer4 – Proširenje baze znanja

- Ako neko vodi zdrav život onda on ne pije.
- $\forall x \text{ VodiZdravŽivot}(x) \Rightarrow \neg \text{Pije}(y) \dots \Delta$
- Ako ljudi smatraju da je neko duhovit onda je on stvarno duhovit.
- $\forall x \forall y (\text{Smatra}(x, y) \Rightarrow \text{DuhovitZa}(y, x)) \Rightarrow \text{Duhovit}(y) \dots \Delta$
- Ako je neko član Mense onda je on inteligentan.
- $\forall x \text{ ČlanMense}(x) \Rightarrow \text{Inteligentan}(x) \dots \Delta$
- Ako ljudi smatraju da je neko zgodan onda je on stvarno zgodan.
- $\forall x \forall y (\text{Smatra}(x, y) \Rightarrow \text{ZgodanZa}(y, x)) \Rightarrow \text{Zgodan}(y) \dots \Delta$
- Milan je momak.
- $\text{Mamak}(\text{Milan}) \dots \Delta$
- Milena je devojka.
- $\text{Devojka}(\text{Milena}) \dots \Delta$



Primer4 – Formiranje u klauzula

1. $\forall x \forall y \text{ Devojka}(x) \wedge \text{Momak}(y) \wedge \neg \text{Pije}(y) \wedge \text{Duhovit}(x) \wedge \text{Pametn}(y) \Rightarrow \text{Voli}(x, y) \dots \Delta$
2. $\forall x \forall y \text{ Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Zgodan}(y) \wedge \text{Intelligentan}(y) \Rightarrow \text{Voli}(x, y) \dots \Delta$
3. $\text{VodiZdravZivot}(\text{Milan}) \wedge \text{Pametn}(\text{Milan}) \dots \Delta$
4. $\forall x \text{ Smatra}(x, \text{Milan}) \Rightarrow \text{DuhovitZa}(\text{Milan}, x) \dots \Delta$
5. $\text{C} \text{lanMense}(\text{Milena}) \dots \Delta$
6. $\forall x \text{ Smatra}(x, \text{Milena}) \Rightarrow \text{ZgodanZa}(\text{Milena}, x) \dots \Delta$
7. $\forall x \forall y \text{ Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Voli}(x, y) \wedge \text{Voli}(y, x) \Rightarrow \text{DobarPar}(x, y) \dots \Delta$
8. $\forall x \text{ VodiZdravZivot}(x) \Rightarrow \neg \text{Pije}(y) \dots \Delta$
9. $\forall x \forall y (\text{Smatra}(x, y) \Rightarrow \text{DuhovitZa}(y, x)) \Rightarrow \text{Duhovit}(y) \dots \Delta$
10. $\forall x \text{ C} \text{lanMense}(x) \Rightarrow \text{Intelligentan}(x) \dots \Delta$
11. $\forall x \forall y (\text{Smatra}(x, y) \Rightarrow \text{ZgodanZa}(y, x)) \Rightarrow \text{Zgodan}(y) \dots \Delta$
12. $\text{Momak}(\text{Milan}) \dots \Delta$
13. $\text{Devojka}(\text{Milena}) \dots \Delta$
14. $\neg \text{DobarPar}(\text{Milan}, \text{Milena}) \dots \Gamma$



Primer4 – Formiranje u klauzula (1)

1. $\forall x \forall y \text{ Devojka}(x) \wedge \text{Mamak}(y) \wedge \neg \text{Pije}(y) \wedge \text{Duhovit}(x) \wedge \text{Pametn}(y) \Rightarrow \text{Voli}(x, y) \dots \Delta$
 $\forall x \forall y \neg(\text{Devojka}(x) \wedge \text{Mamak}(y) \wedge \neg \text{Pije}(y) \wedge \text{Duhovit}(x) \wedge \text{Pametn}(y)) \vee \text{Voli}(x, y) \dots \Delta$
2. $\forall x \forall y \text{ Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Zgodan}(y) \wedge \text{Inteligentan}(y) \Rightarrow \text{Voli}(x, y) \dots \Delta$
 $\forall x \forall y \neg(\text{Mamak}(x) \wedge \text{Devojka}(y) \wedge \text{Zgodan}(y) \wedge \text{Inteligentan}(y)) \vee \text{Voli}(x, y) \dots \Delta$
4. $\forall x \text{ Smatra}(x, \text{Milan}) \Rightarrow \text{DuhovitZa}(\text{Milan}, x) \dots \Delta$
 $\forall x \neg \text{Smatra}(x, \text{Milan}) \vee \text{DuhovitZa}(\text{Milan}, x) \dots \Delta$
6. $\forall x \text{ Smatra}(x, \text{Milena}) \Rightarrow \text{ZgodanZa}(\text{Milena}, x) \dots \Delta$
 $\forall x \neg \text{Smatra}(x, \text{Milena}) \vee \text{ZgodanZa}(\text{Milena}, x) \dots \Delta$
7. $\forall x \forall y \text{ Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Voli}(x, y) \wedge \text{Voli}(y, x) \Rightarrow \text{DobarPar}(x, y) \dots \Delta$
 $\forall x \forall y \neg(\text{Mamak}(x) \wedge \text{Devojka}(y) \wedge \text{Voli}(x, y) \wedge \text{Voli}(y, x)) \vee \text{DobarPar}(x, y) \dots \Delta$
8. $\forall x \text{ VodiZdravZivot}(x) \Rightarrow \neg \text{Pije}(y) \dots \Delta$
 $\forall x \neg \text{VodiZdravZivot}(x) \vee \neg \text{Pije}(y) \dots \Delta$
9. $\forall x \forall y (\text{Smatra}(x, y) \Rightarrow \text{DuhovitZa}(y, x)) \Rightarrow \text{Duhovit}(y) \dots \Delta$
 $\forall x \forall y \neg(\neg \text{Smatra}(x, y) \vee \text{DuhovitZa}(y, x)) \vee \text{Duhovit}(y) \dots \Delta$
10. $\forall x \text{ ČlanMense}(x) \Rightarrow \text{Inteligentan}(x) \dots \Delta$
 $\forall x \neg \text{ČlanMense}(x) \vee \text{Inteligentan}(x) \dots \Delta$
11. $\forall x \forall y (\text{Smatra}(x, y) \Rightarrow \text{ZgodanZa}(y, x)) \Rightarrow \text{Zgodan}(y) \dots \Delta$
 $\forall x \forall y \neg(\neg \text{Smatra}(x, y) \vee \text{ZgodanZa}(y, x)) \vee \text{Zgodan}(y) \dots \Delta$



Primer4 – Formiranje u klauzula (2)

1. $\forall x \forall y \neg(\text{Devojka}(x) \wedge \text{Momak}(y) \wedge \neg \text{Pije}(y) \wedge \text{Duhovit}(x) \wedge \text{Pametn}(y)) \vee \text{Voli}(x, y) \dots \Delta$
 $\forall x \forall y \neg \text{Devojka}(x) \vee \neg \text{Momak}(y) \vee \text{Pije}(y) \vee \neg \text{Duhovit}(x) \vee \neg \text{Pametn}(y) \vee \text{Voli}(x, y) \dots \Delta$
2. $\forall x \forall y \neg \text{Momak}(x) \vee \neg \text{Devojka}(y) \vee \neg \text{Zgodan}(y) \vee \neg \text{Intelligentan}(y) \vee \text{Voli}(x, y) \dots \Delta$
 $\forall x \forall y \neg \text{Momak}(x) \vee \neg \text{Devojka}(y) \vee \neg \text{Zgodan}(y) \vee \neg \text{Intelligentan}(y) \vee \text{Voli}(x, y) \dots \Delta$
7. $\forall x \forall y \neg(\text{Momak}(x) \wedge \text{Devojka}(y) \wedge \text{Voli}(x, y) \wedge \text{Voli}(y, x)) \vee \text{DobarPar}(x, y) \dots \Delta$
 $\forall x \forall y \neg \text{Momak}(x) \vee \neg \text{Devojka}(y) \vee \neg \text{Voli}(x, y) \vee \neg \text{Voli}(y, x) \vee \text{DobarPar}(x, y) \dots \Delta$
9. $\forall x \forall y \neg(\neg \text{Smatra}(x, y) \vee \text{DuhovitZa}(y, x)) \vee \text{Duhovit}(y) \dots \Delta$
 $\forall x \forall y \text{Smatra}(x, y) \wedge \neg \text{DuhovitZa}(y, x) \vee \text{Duhovit}(y) \dots \Delta$
11. $\forall x \forall y \neg(\neg \text{Smatra}(x, y) \vee \text{ZgodanZa}(y, x)) \vee \text{Zgodan}(y) \dots \Delta$
 $\forall x \forall y \text{Smatra}(x, y) \wedge \neg \text{ZgodanZa}(y, x) \vee \text{Zgodan}(y) \dots \Delta$



Primer4 – Formiranje u klauzula (5)

1. $\neg \text{Devojka}(x) \vee \neg \text{Momak}(y) \vee \text{Pije}(y) \vee \neg \text{Duhovit}(x) \vee \neg \text{Pametn}(y) \vee \text{Voli}(x, y) \dots \Delta$
2. $\neg \text{Momak}(x) \vee \neg \text{Devojka}(y) \vee \neg \text{Zgodan}(y) \vee \neg \text{Intelligentan}(y) \vee \text{Voli}(x, y) \dots \Delta$
3. $\text{VodiZdravZivot}(\text{Milan}) \wedge \text{Pametn}(\text{Milan}) \dots \Delta$
4. $\neg \text{Smatra}(x, \text{Milan}) \vee \text{DuhovitZa}(\text{Milan}, x) \dots \Delta$
5. $\text{C}lanMense(\text{Milena}) \dots \Delta$
6. $\text{Smatra}(x, \text{Milena}) \wedge \text{ZgodanZa}(\text{Milena}, x) \dots \Delta$
7. $\neg \text{Momak}(x) \vee \neg \text{Devojka}(y) \vee \neg \text{Voli}(x, y) \vee \neg \text{Voli}(y, x) \vee \text{DobarPar}(x, y) \dots \Delta$
8. $\neg \text{VodiZdravZivot}(x) \vee \neg \text{Pije}(y) \dots \Delta$
9. $\text{Smatra}(x, y) \wedge \neg \text{DuhovitZa}(y, x) \vee \text{Duhovit}(y) \dots \Delta$
10. $\neg \text{C}lanMense(x) \vee \text{Intelligentan}(x) \dots \Delta$
11. $\text{Smatra}(x, y) \wedge \neg \text{ZgodanZa}(y, x) \vee \text{Zgodan}(y) \dots \Delta$
12. $\text{Momak}(\text{Milan}) \dots \Delta$
13. $\text{Devojka}(\text{Milena}) \dots \Delta$
14. $\neg \text{DobarPar}(\text{Milan}, \text{Milena}) \dots \Gamma$



Primer4 – Formiranje u klauzula (7)

1. $\{\neg \text{Devojka}(x), \neg \text{Momak}(y), \text{Pije}(y), \neg \text{Duhovit}(x), \neg \text{Pametn}(y), \text{Voli}(x, y)\} \dots \Delta$
2. $\{\neg \text{Momak}(x), \neg \text{Devojka}(y), \neg \text{Zgodan}(y), \neg \text{Inteligentan}(y), \text{Voli}(x, y)\} \dots \Delta$
3. $\{\text{VodiZdravZivot}(\text{Milan})\} \quad \{\text{Pametn}(\text{Milan})\} \dots \Delta$
4. $\{\neg \text{Smatra}(x, \text{Milan}), \text{DuhovitZa}(\text{Milan}, x)\} \dots \Delta$
5. $\{\text{ČlanMense}(\text{Milena})\} \dots \Delta$
6. $\{\neg \text{Smatra}(x, \text{Milena}), \text{ZgodanZa}(\text{Milena}, x)\} \dots \Delta$
7. $\{\neg \text{Momak}(x), \neg \text{Devojka}(y), \neg \text{Voli}(x, y), \neg \text{Voli}(y, x), \text{DobarPar}(x, y)\} \dots \Delta$
8. $\{\neg \text{VodiZdravZivot}(x), \neg \text{Pije}(y)\} \dots \Delta$
9. $\{\text{Smatra}(x, y), \text{Duhovit}(y)\} \quad \{\neg \text{DuhovitZa}(y, x), \text{Duhovit}(y)\} \dots \Delta$
10. $\{\neg \text{ČlanMense}(x), \text{Inteligentan}(x)\} \dots \Delta$
11. $\{\text{Smatra}(x, y), \text{Zgodan}(y)\} \quad \{\neg \text{ZgodanZa}(y, x), \text{Zgodan}(y)\} \dots \Delta$
12. $\{\text{Momak}(\text{Milan})\} \dots \Delta$
13. $\{\text{Devojka}(\text{Milena})\} \dots \Delta$
14. $\{\neg \text{DobarPar}(\text{Milan}, \text{Milena})\} \dots \Gamma$



Primer4 – Formiranje u klauzula (8)

1. $\{\neg \text{Devojka}(a), \neg \text{Momak}(b), \text{Pije}(b), \neg \text{Duhovit}(b), \neg \text{Pametan}(b), \text{Voli}(a, b)\}$
2. $\{\neg \text{Momak}(c), \neg \text{Devojka}(d), \neg \text{Zgodan}(d), \neg \text{Intelligentan}(d), \text{Voli}(c, d)\}$
3. $\{\text{VodiZdravZivot}(\text{Milan})\}$
4. $\{\text{Pametan}(\text{Milan})\}$
5. $\{\neg \text{Smatra}(e, \text{Milan}), \text{DuhovitZa}(\text{Milan}, e)\}$
6. $\{\text{ČlanMense}(\text{Milena})\}$
7. $\{\neg \text{Smatra}(f, \text{Milena}), \text{ZgodanZa}(\text{Milena}, f)\}$
8. $\{\neg \text{Momak}(g), \neg \text{Devojka}(h), \neg \text{Voli}(g, h), \neg \text{Voli}(h, g), \text{DobarPar}(g, h)\}$
9. $\{\neg \text{VodiZdravZivot}(i), \neg \text{Pije}(i)\}$
10. $\{\text{Smatra}(j, k), \text{Duhovit}(k)\}$
11. $\{\neg \text{DuhovitZa}(l, m), \text{Duhovit}(l)\}$
12. $\{\neg \text{ČlanMense}(n), \text{Intelligentan}(n)\}$
13. $\{\text{Smatra}(o, p), \text{Zgodan}(p)\}$
14. $\{\neg \text{ZgodanZa}(q, r), \text{Zgodan}(q)\}$
15. $\{\text{Momak}(\text{Milan})\}$
16. $\{\text{Devojka}(\text{Milena})\}$
17. $\{\neg \text{DobarPar}(\text{Milan}, \text{Milena})\}$



Primer4 – Zaključivanje(1)

7. $\{\neg \text{Smatra}(f, \text{Milena}), \text{ZgodanZa}(\text{Milena}, f)\}$

13. $\{\text{Smatra}(o, p), \text{Zgodan}(p)\}$

• Iz 7. i 13. unifikacijom promenljivih $\{f/s, o/s, p/\text{Milena}\}$ dobija se:

18. $\{\text{ZgodanZa}(\text{Milena}, s), \text{Zgodan}(\text{Milena})\}$

14. $\{\neg \text{ZgodanZa}(q, r), \text{Zgodan}(q)\}$

18. $\{\text{ZgodanZa}(\text{Milena}, s), \text{Zgodan}(\text{Milena})\}$

• Iz 14. i 18. unifikacijom promenljivih $\{s/t, r/t, q/\text{Milena}\}$ dobija se:

19. $\{\text{Zgodan}(\text{Milena})\}$

6. $\{\text{ČlanMense}(\text{Milena})\}$

12. $\{\neg \text{ČlanMense}(n), \text{Intelligentan}(n)\}$

• Iz 6. i 12. unifikacijom promenljivih $\{n/\text{Milena}\}$ dobija se:

20. $\{\text{Intelligentan}(\text{Milena})\}$



Primer4 – Zaključivanje(2)

15. {Mamak(Milan)}

16. {Devojka(Milena)}

19. {Zgodan(Milena)}

20. {Intelligentan(Milena)}

2. { \neg Mamak(c), \neg Devojka(d), \neg Zgodan(d), \neg Intelligentan(d), Voli(c, d)}

- Iz 15., 16., 19., 20. i 2. unifikacijom promenljivih **{c/Milan, d/Milena}** dobija se:

21. {Voli(Milan, Milena)}



Primer4 – Zaključivanje(3)

3. {VodiZdravŽivot(Milan)}

9. {¬VodiZdravŽivot(i), ¬PiJe(i)}

• Iz 3. i 9. unifikacijom promenljivih {i/Milan} dobija se:

22. {¬PiJe(Milan)}

5. {¬Smatra(e, Milan), DuhovitZa(Milan, e)}

10. {Smatra(j, k), Duhovit(k)}

• Iz 5. i 10. unifikacijom promenljivih {e/u, j/u, k/Milan} dobija se:

23. {DuhovitZa(Milan, u), Duhovit(Milan)}

11. {¬DuhovitZa(l, m), Duhovit(l)}

23. {DuhovitZa(Milan, u), Duhovit(Milan)}

• Iz 11. i 23. unifikacijom promenljivih {m/v, u/v, l/Milan} dobija se:

24. {Duhovit(Milan)}



Primer4 – Zaključivanje(4)

16. {Devojka(Milena)}

15. {Momak(Milan)}

22. {¬Pije(Milan)}

24. {Duhovit(Milan)}

4. {Pametna(Milana)}

1. {¬Devojka(a), ¬Momak(b), Pije(b),
¬Duhovit(b), ¬Pametna(b), Voli(a, b)}

- Iz 16., 15., 22., 24., 4. i 1. unifikacijom promenljivih {a/Milena, b/Milan} dobija se:

25. {Voli(Milena, Milan)}



Primer4 – Zaključivanje(5)

15. {Momak(Milan)}

16. {Devojka(Milena)}

21. {Voli(Milan, Milena)}

25. {Voli(Milena, Milan)}

17. {¬DobarPar(Milan, Milena)}

8. {¬Momak(g), ¬Devojka(h), ¬Voli(g, h),
¬Voli(h, g), DobarPar(g, h)}

- Iz 15., 16., 21., 25., 17. i 8. unifikacijom promenljivih {g/Milan, h/Milena} dobija se:

26. { }

