



# Ekspertski sistemi

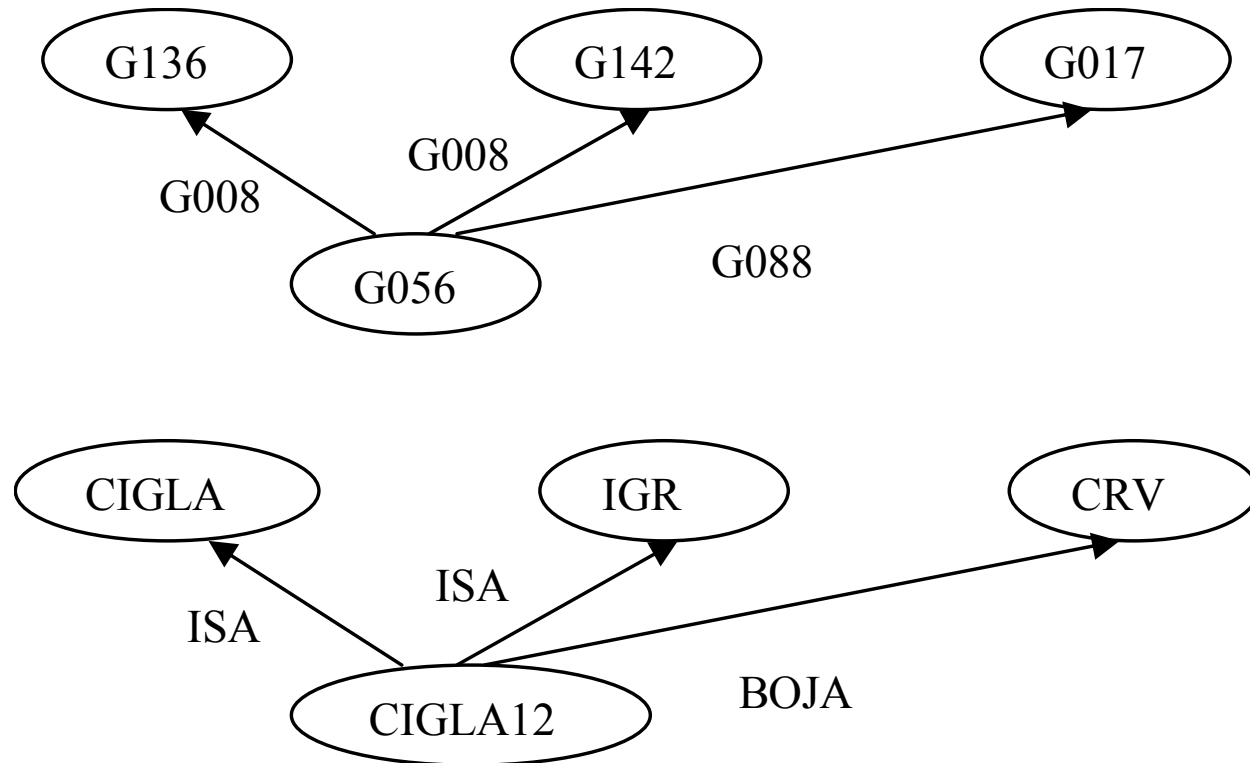
## Lekcija 5: PREDSTAVLJANJE NEFORMALNOG ZNANJA

# Semantičke mreže

- Semantičke mreže opisuju element i veze između njih.
- Objekti se označavaju sa  i 
- veze se označenim strelicama.
- Elementi su čvorovi, a strelice su veze.

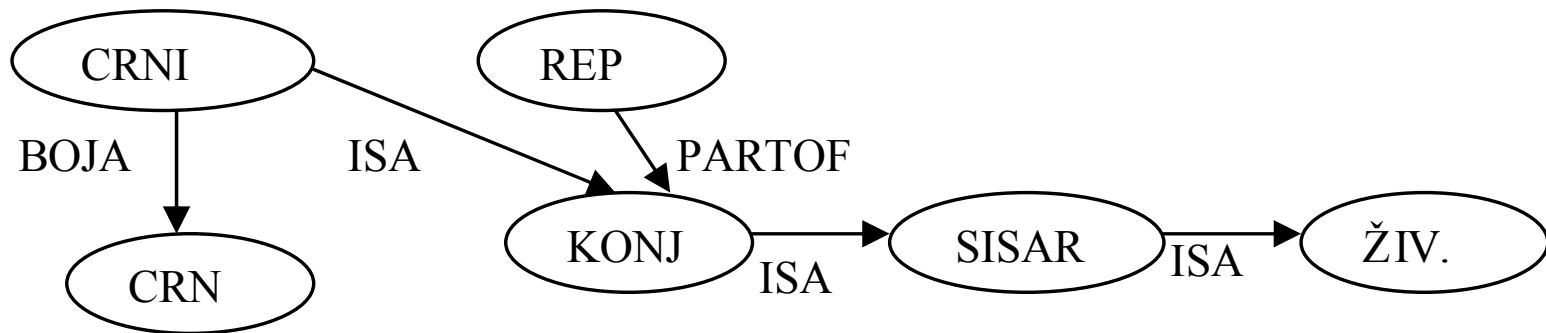
# Semantičke mreže

- Mora se uključiti i semantika



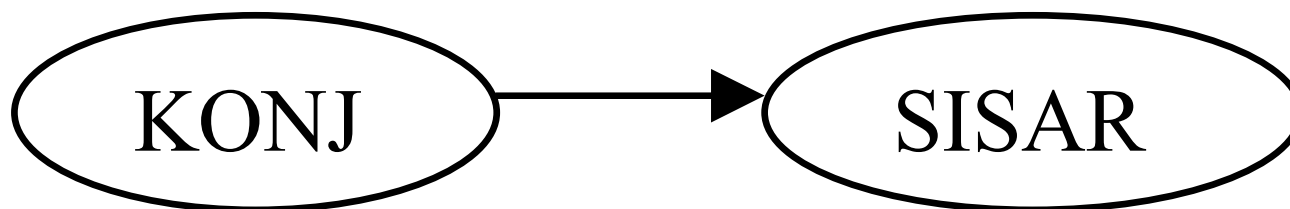
# Semantičke mreže

- Veza ISA se koristi da označi pripadnost nekog elementa klasi elemenata.
- PARTOF se koristi da označi da je neki element deo nekog elementa.



# Semantičke mreže

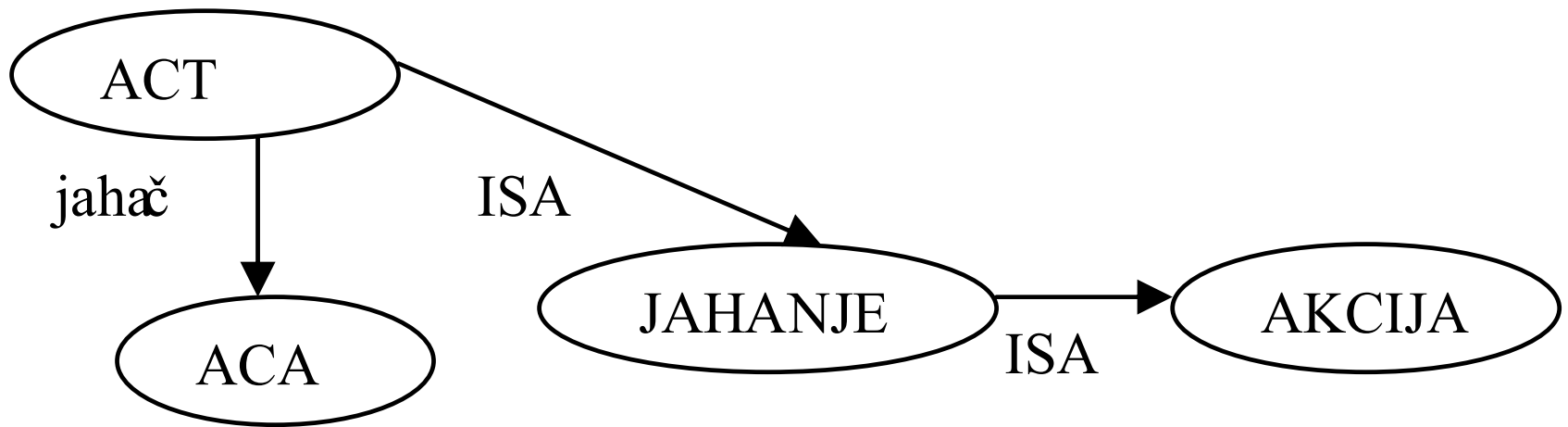
- Čvor koji predstavlja specifičan primer klase naziva se primerak (instanca).



- **Pregradak** čvora čine veze različitih imena.
- Za primer sa ciglom dva pregradka: jedan pregradak sa dve vrednosti (CIGLA, IGRAČKA), i drugi pregradak sa jednom vrednošću (CRVENA).

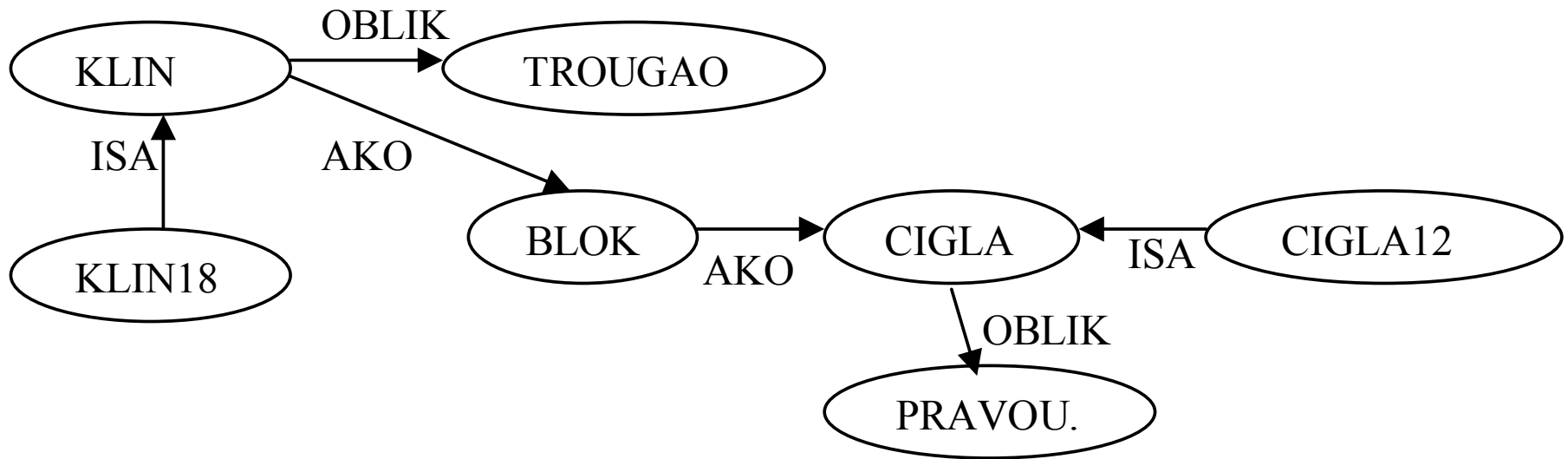
# Semantičke mreže

- Koncept klase i ISA veza se može iskoristiti da se predstavi situacija, akcija i događaj



# Semantičke mreže

- Nasleđivanje omogućava pomeranje opisa sa klase na instance (uzorke).



# Semantičke mreže

- Procedura nasleđivanja vrednosti glasi:

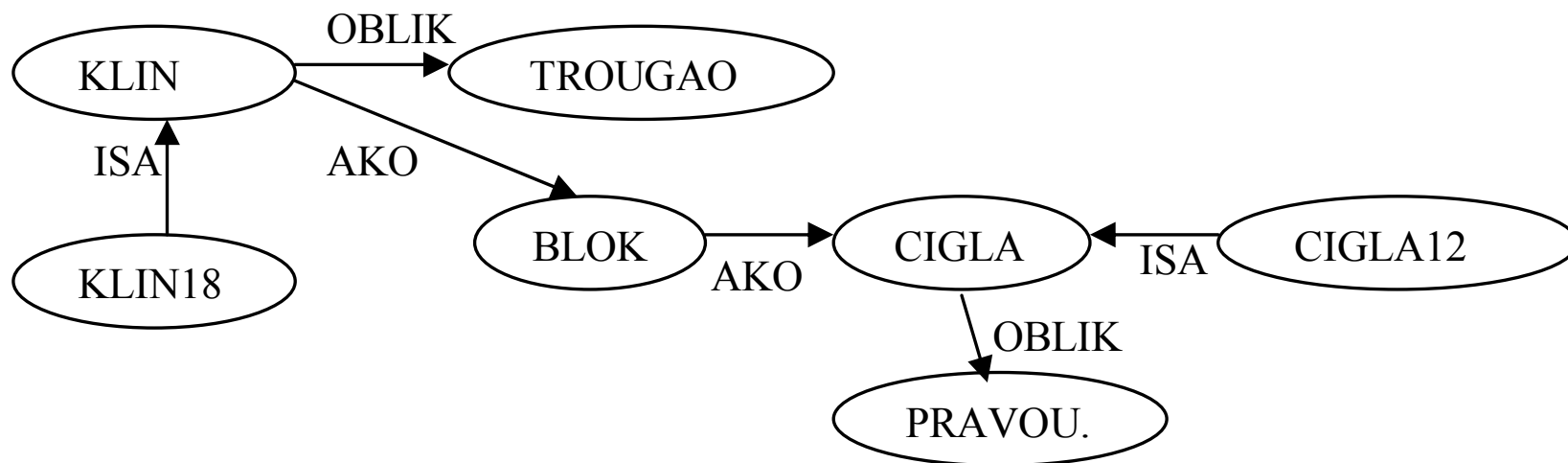
Neka je F dati čvor, a S dati pregradak:

- sačiniti red čekanja koji se sastoji od čvora F i svih ostalih čvorova klase koji se nalaze u ISA pregradku čvora F; F treba da bude na čelu;
- dok red čekanja ne postane prazan ili dok se ne nađe vrednost, odrediti da li prvi element reda čekanja ima vrednost u svom S pregradku:
  - ako prvi element ima vrednost u svom S pregradku, tada je vrednost pronađena,
  - inače ukloniti prvi element iz reda čekanja i dodati čvorove vezane za prvi element u ovom trenutku sa AKO na kraj reda čekanja
- ako je vrednost pronađena obavestiti da je pronađena vrednost vrednost pregradka S čvora F, inače obavestiti o neuspehu.



# Semantičke mreže

- Procedura nasleđivanja - primer sa elementom KLIN18:
  1. Red čekanje: KLIN18 (F, polazni čvor), KLIN (ISA pregradak čvora F)  
S pregradak: OBLIK
  2. KLIN18 - nema S pregradaka
  3. KLIN BLOK
  2. KLIN - ima S pregradak



# Semantičke mreže

- Nekada ne postoje vrednosti u pregradkama, već se mogu izračunati koristeći postojeće informacije.
- Na primer, može se izračunati težina bloka iz podataka o zapremini i materijalu.
- Koriste se procedure koje se nazivaju if-needed procedure, jer se koriste kada je potrebno, a ne kada se posebno zatraži njihova usluga pozivanjem po imenu.

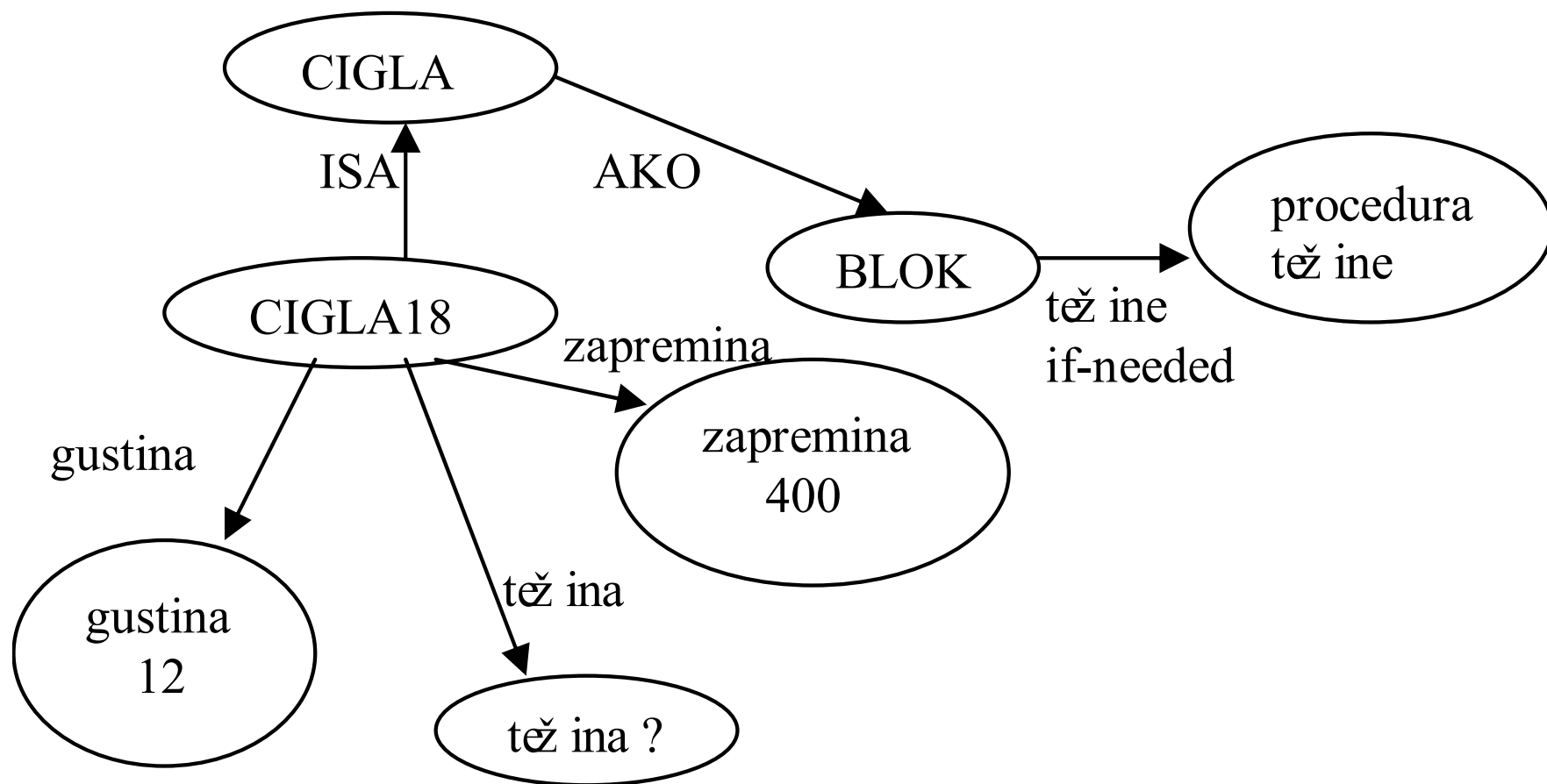
# Semantičke mreže

- Nasleđivanje if-needed procedura:

Neka je dat čvor  $F$  i njegov pregradak  $S$ . Ukoliko  $S$  ne sadrži nikakvu vrednost (prazan je), uraditi sledeće:

- formirati red čekanja koji se sastoji od čvora  $F$  i svih čvorovaklase povezanih sa čvorom  $F$  preko ISA pregradka; čvor  $F$  treba da bude na čelu;
- Dok red čekanja ne postane prazan ili je uspešna if-needed procedura pronađena, odrediti da li prvi element reda čekanja ima proceduru u if-needed delu svog pregradka:
  - ako postoji procedura, i ako procedura daje vrednost, tada je vrednost pronađena,
  - inače, ukloniti prvi element iz reda čekanja i dodati čvorovevezane sa prvim elementom sa AKO na kraju reda čekanja
- ako procedura nađe vrednost, reći da je nađena vrednost vrednost  $S$  pregradka čvora  $F$ , inače obavestiti i neuspehu.

# Semantičke mreže



# Semantičke mreže

Izvršavanje procedure je:

1. Red čekanja: CIGLA18 (F - početni čvor), CIGLA

Pregradak S: težina ?

2. Ne postoji procedura if-needed u okviru elementa CIGLA18

Red čekanja: CIGLA, BLOK

3. Ne postoji procedura if-needed u okviru elementa CIGLA

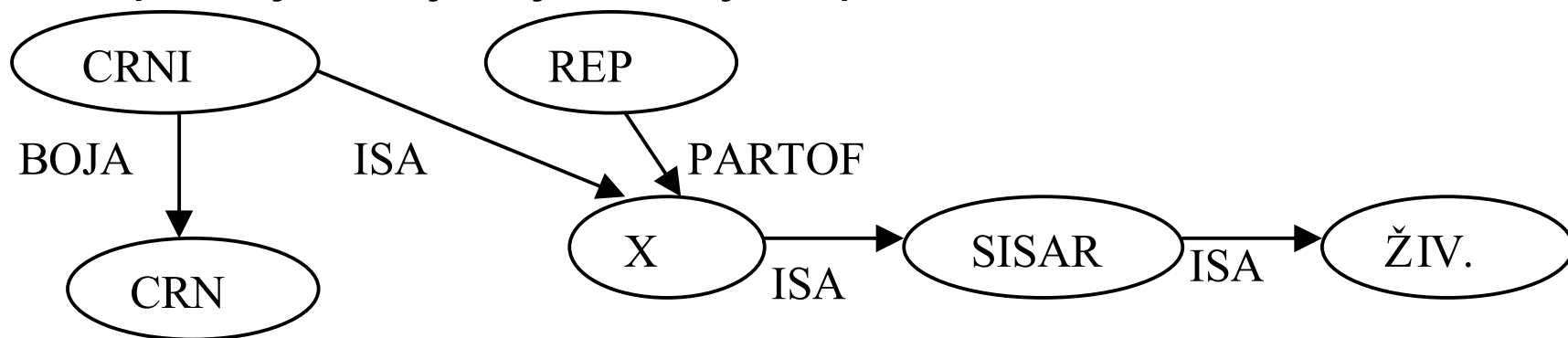
Red čekanja: BLOK

4. Izračunavanje težine

5. Pregradak S: težina = 4800

# Rezonovanje i semantičke mreže

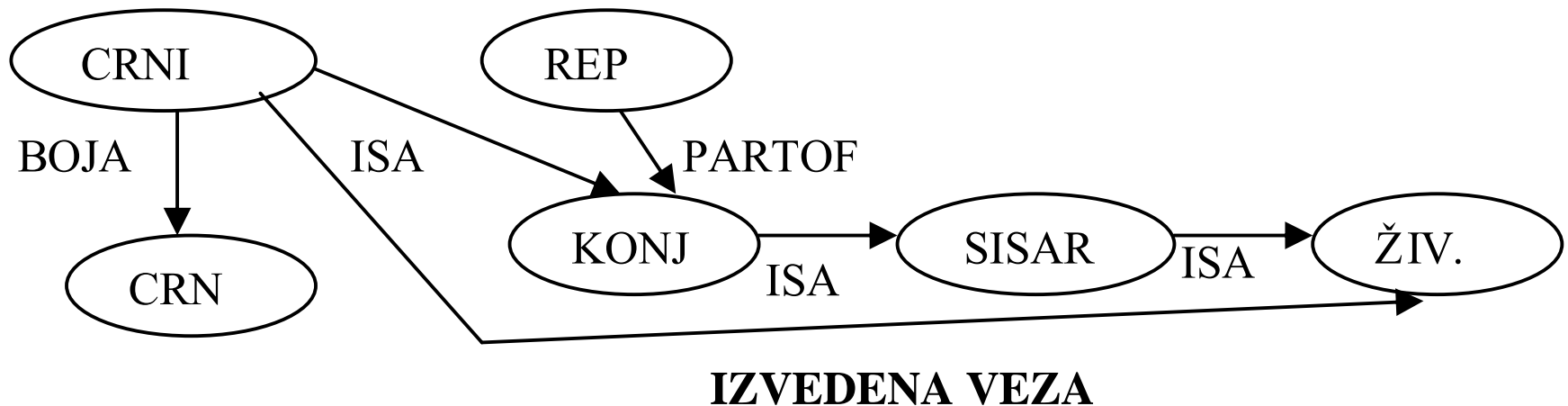
- Rezonovanje se sastoji u praćenju veza u mreži.
- Zaključuje se na bazi veza i generišu se nove veze.
- Može se desiti da se prostim sledom dođe do zaključka koji nisu potpuno tačno zbog neprepoznavanja izuzetaka - postoje konji koji nemaju repove.



- Konstruiše se upit u vidu mreže, pa se vrši traganje u postojećoj semantičkoj mreži koja će potpuno da se upari sa upitom. Zatim se nepoznata vrednost čvora vezuje za vrednost čvora u semantičkoj mreži. Znači  $X \rightarrow \text{KONJ}$

# Rezonovanje i semantičke mreže

- Primer kako se vrši uparivanje sa stvarima koje imaju delove, na osnovu izvođenja novih veza iz postojećih.
- Na primer, koja životinja je crni konj sa repom.



# OKVIRI

- Okviri su jedan od načina predstavljanja znanja kojim se organizuje predstavljanje istinitih stvari za neku opštu klasu elemenata.
- Okviri su u stvari skup semantičkih čvorova i pregradaka koji opisuju zajedno stereotipske objekte, događaje, akcije.
- Nekada se razvijaju mreže u kojima su čvorovi okviri.
- Kod okvira je naglasak na implicitnom znanju.



# OKVIRI

- Okviri se sastoje od pregradaka od kojih svaki predstavlja standardno svojstvo ili atribut elementa koji je predstavljen okvirom.
- Pregradak omogućava način da se sistematski smesti jedna komponenta stečenog iskustva u odnosu na predstavljenu klasu elemenata.
- I sam pregradak može da predstavlja etalon po kome se pamte instance, t.j. jednu relaciju atributa.

# OKVIRI

Frame: ZEMLJOTRES

specialization: DISASTER-EVENT

poremećaji: n.a.

jačina: n.a.

Frame: DISASTER-EVENT

specialization: EVENT

poginuli: n.a.

povređeni: n.a.

broj uništenih kuća: n.a.

materijalna šteta: n.a.

# OKVIRI

Frame: EVENT

mesto: n.a.

datum: n.a.

vreme: n.a.

Frame: URAGANSKI\_VETROVI

specialaization: DISASTER-EVENT

ime: n.a.

mesto\_pojavljivanja: n.a.

brzina\_vetra: n.a.

# OKVIRI

Desio se zemljotres na primer - novi okvir kao instanca ove klase. Na primer:

Frame: ZEMLJOTRES18

mesto: SKOPLJE

datum: 26.7.1963.

vreme: 5:10

poginuli: 1200

povređeni: 4500

broj uništenih kuća: 10000

materijalna šteta: 300 M\$

poremećaji: srednji

jačina: 6.5 R

# OKVIRI

- Okviri koji predstavljaju klasu objekta na datom nivou apstrakcije mogu nasleđivati pregrade i vrednosti koje se nalaze na višem nivou apstrakcije.
- Proceduralne informacije koje se ugrađuju su:
  - if-needed procedura, ostvaruje se kada treba da izračuna vrednost pregradka
  - if-addes procedura, aktivira se kada je vrednost pregradka već dodeljena.

# OKVIRI

Frame kola

specijalizacija: drumsko\_vozilo

telo: čelik

prozori: staklo

preostalo\_gorivo:

opseg: (prazana, četvrtina, pola, pun)

implicit: nema

if-needed: proveriti\_pokazivač\_goriva

vrsta\_oštećenja:

implicit: nema

if-added: pozvati\_osiguravajući\_zavod

end\_frame

# Sistemi zasnovani na okvirima

- Računarski program koji obrađuje specifične informacije problema, koje se nalaze u radnoj memoriji skupom okvira iz baze znanja, koristeći mašinu za zaključivanje da bi se izvele nove informacije naziva se sistem zasnovan na okvirima.

klasa:

ljudska  
bića

svojstva

ime

vrednost

aspekti

podklasa:

muškarci

žene

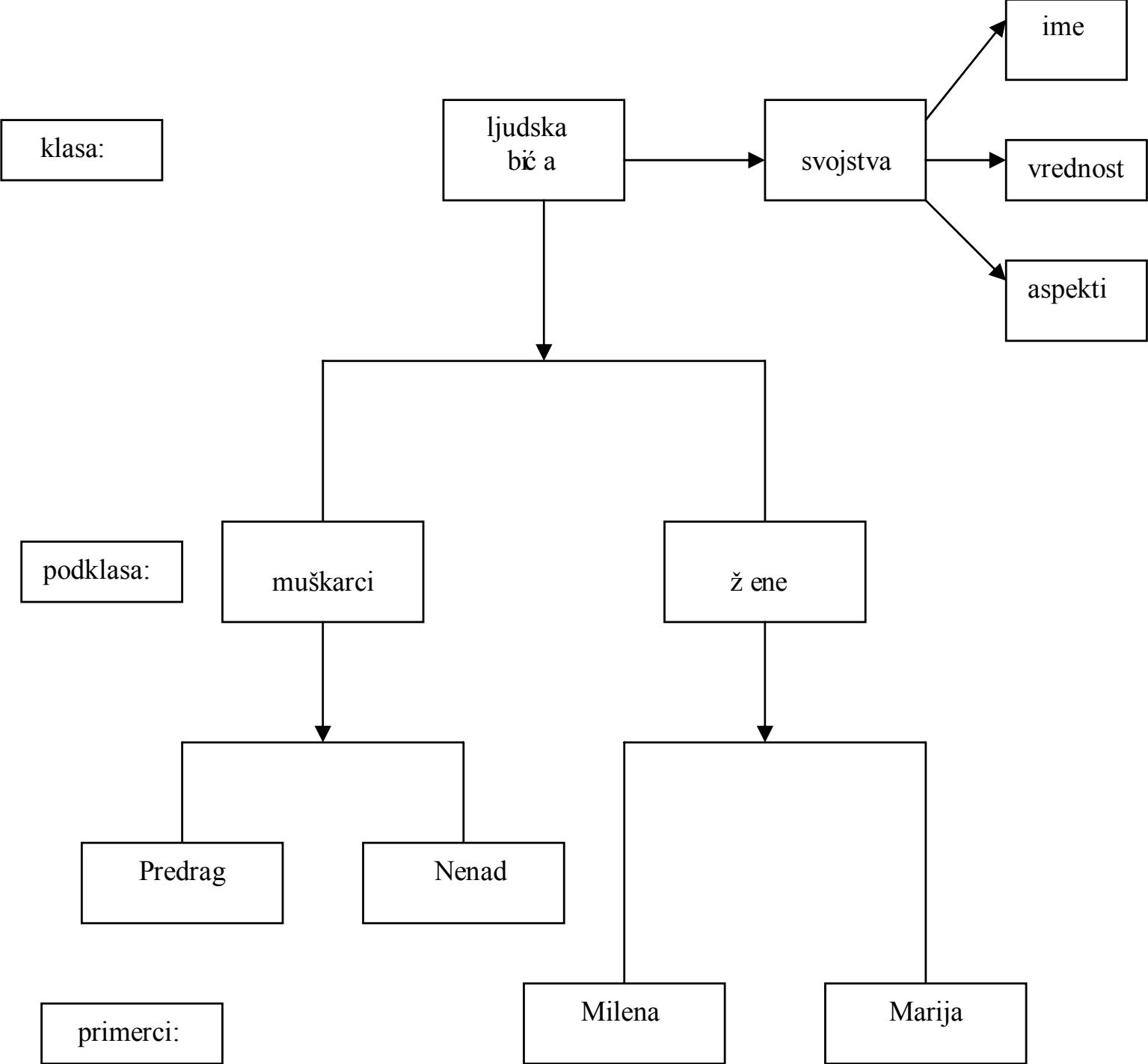
Predrag

Nenad

primeri:

Milena

Marija





# OKVIRI

- **Ciljna agenda:** obezbeđuje listu zadataka koje treba obaviti, slično kao u sistemima zasnovanim na pravilima.
- **Skup pravila:** uključuje moćna pravila uparivanja uzoraka koji omogućavaju izvlačenje zaključaka iz celokupnog sveta okvira pregledanjem svih okvira u potrazi za informacijama koje podržavaju zaključke.
- **Okvir klasa:** sadrži generičke informacije o konceptu, a to su: deskriptivno ime koncepta, skup osobina koje su karakteristične za sve pridružene objekte, kao i vrednost osobina za koje se smatra da su zajedničke objektima.

# OKVIRI – klasa ljudska bića

ime klase:

ljudska bića

podklase:

muškarci, žene

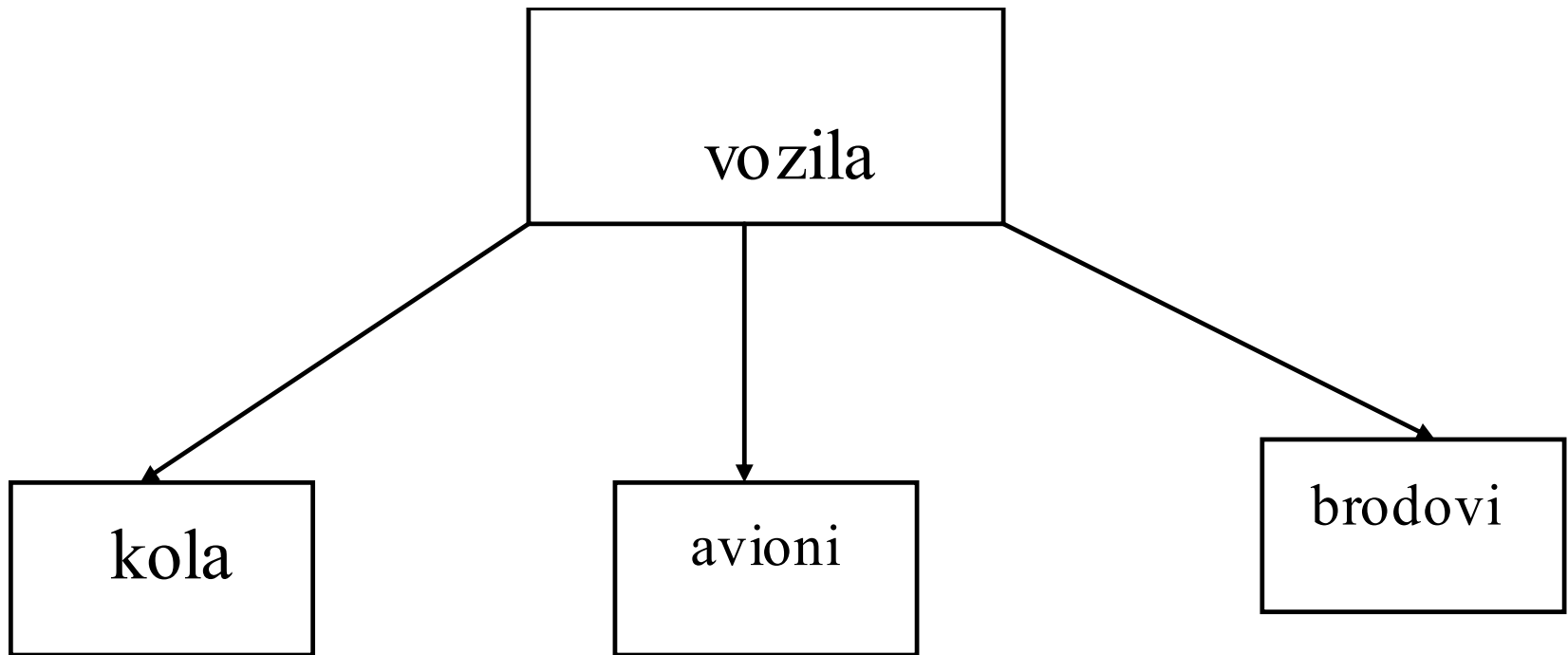
svojstva

starost	?
broj nogu	2
boravište	?
čekivana dužina života	72

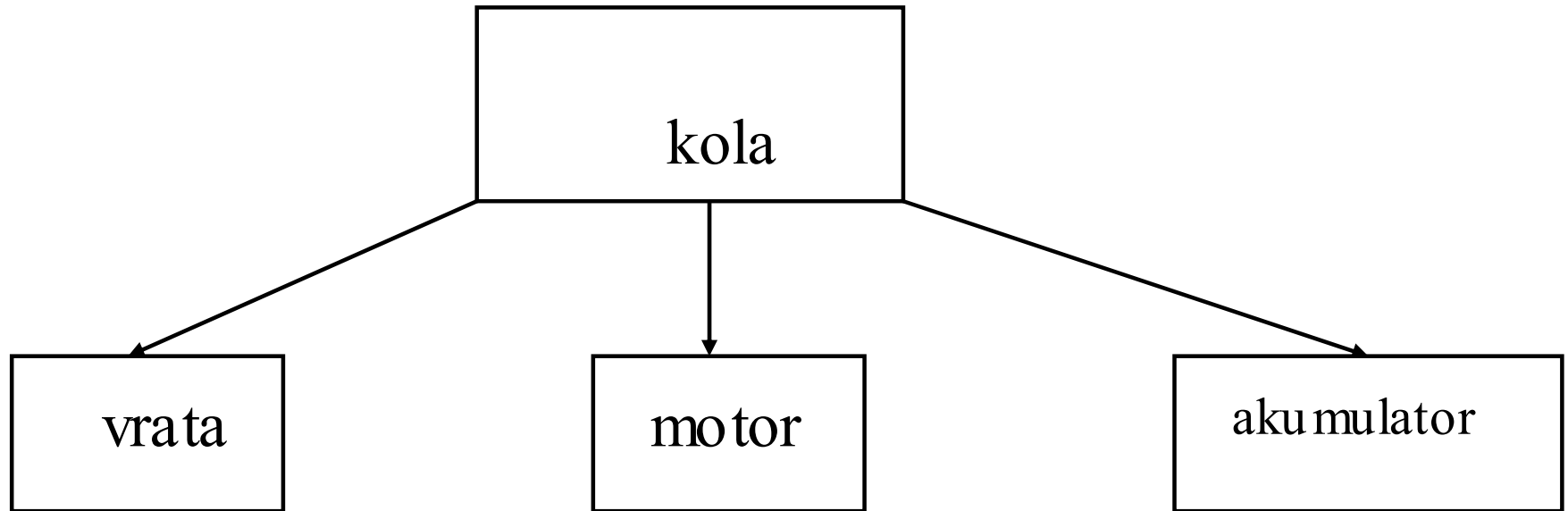
# OKVIRI

- Podklase se (muškaraca i žena) mogu automatski formirati.
- Svaka podklasa nasleđuje osobine i vrednosti osobina roditelja.
- Postoje tri vrste veza između klasa u slučaju da se koriste podklase:
  - generalizacija – a-kind-of,
  - agregacija – a-part-of,
  - asocijacija – semantic

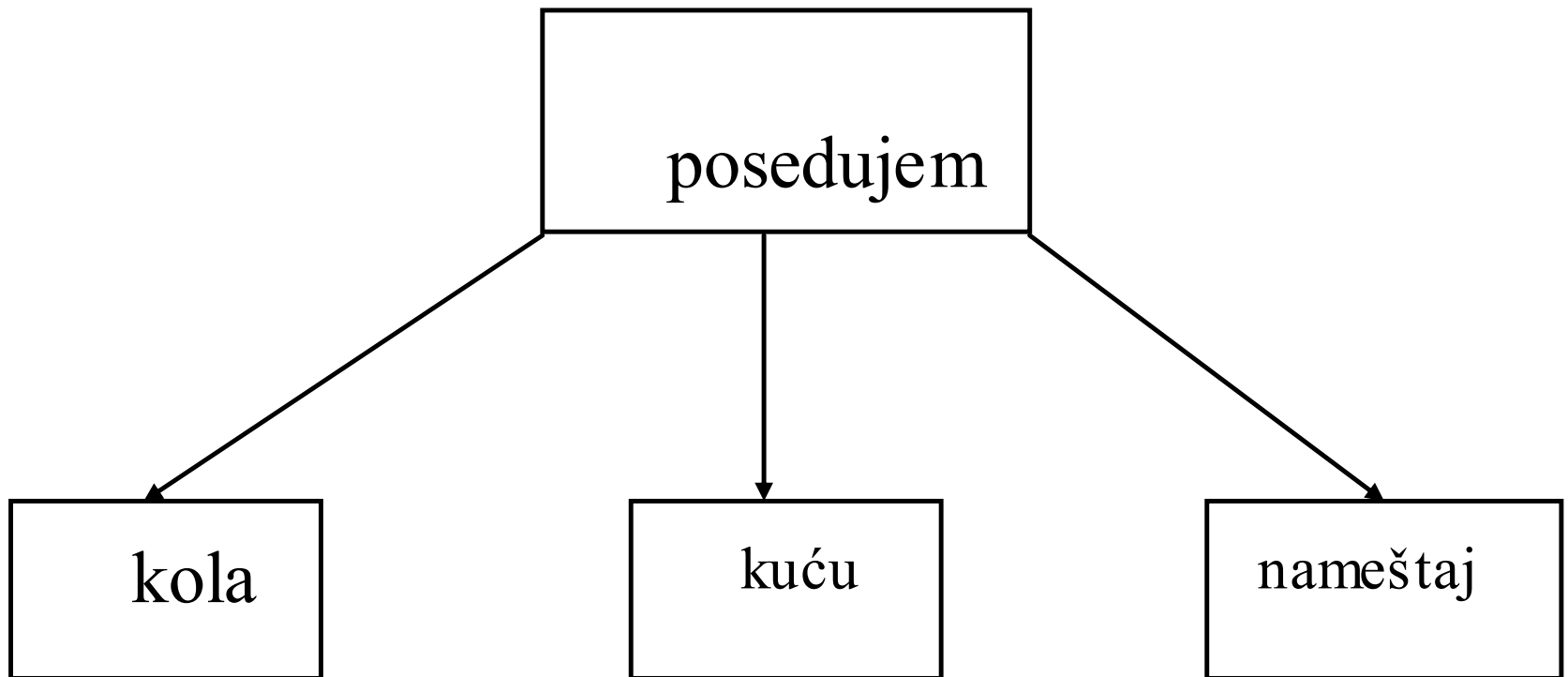
# Generalizacija



# Agregacija



# Asocijacija



# Generalizacija - primer

- Muškarci i žene kao podklase imaju vezu generalizacija prema višoj klasi.
- Klasa muškarci ima sada okvir u sledećem obliku:

ime klase:

muškarci
----------

podklase:

--

svojstva:

starost	?
broj nogu	2
boravište	?
<i>očekivana dužina života</i>	69
<i>brkovi</i>	?

# Nasleđivanje

- Podklase nasleđuju ponašanje svojih roditelja ili nadklasa. Ako je bilo koja procedura priključena aspektu nadklase, ona će biti nasleđena od podklase.
- Vrednosti osobina su podaci koji se dodeljuju podklasi - numeričke, simboličke ili Bulove. Mogu se inicijalizovati pri formiranju okvira ili biti dodeljene u logiku rada ekspertskog sistema.
- Mogu se koristiti i menjati preko pravila ili okvira koji radi sa podacima iz sesije.



# Facets

- **Facets** - obezbeđuje korisniku dodatnu snagu nad tim kako se vrednosti osobina (atributa) uspostavljaju i kontrolišu.
- Instanca ili primerak je specifičan objekat iz klase objekata koji sadrži sve karakteristike okvira klase, ali obično sadrži i informacije koje su specifične za taj objekat.

# Instanca - primer

ime objekta:

Jovan

klasa:

muškarci

svojstva:

starost	18
broj nogu	2
boravište	Beograd
očekivana dužina života	110
brkovi	ne

# Instanca - primer

ime objekta:

hladnjak

klasa:

domaći aparati

svojstva:

starost	10	<i>klasifikacija</i>
servisni broj	125-1995	<i>baza podataka</i>
zapremina	300	<i>procedura</i>
stanje	dobro	<b><i>ekspertski sistem</i></b>
mesto	kuhinja	<i>korisnik</i>
napajanje	električno	<i>nasleđeno</i>
ciklus odmrzavanja	uključen	<i>od drugog objekta</i>

# Nasleđivanje

- Nasleđivanje je proces po kome se karakteristike roditeljskog okvira preuzimaju od strane okvira potomka
- Može se formirati okvir klase koji sadrži sve generičke karakteristike neke klase objekata, a zatim formirati proizvoljan broj primeraka bez eksplicitnog kodiranja karakteristika nivoa klasa (broj nogu,...)
- Pri projektovanju sistema označavajući okvir kao primerak neke klase, sve informacije o klasi se automatski nasleđuju od primeraka, uklanjajući potrebu da se te informacije eksplicitno kodiraju.
- Primerku se mogu dodeliti svojstva, vrednosti i facets koje samo on poseduje

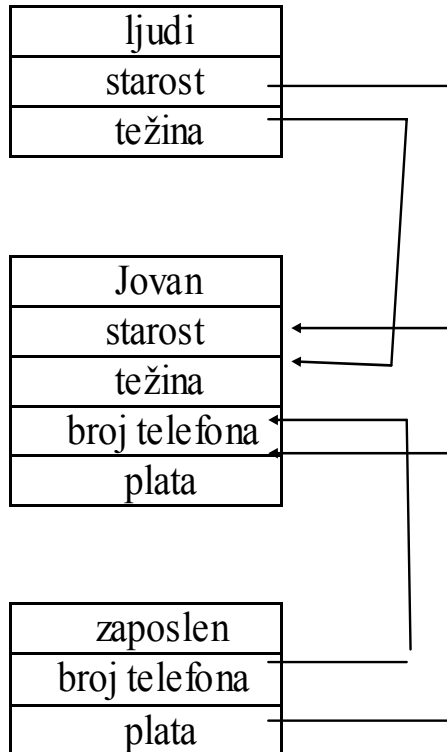
# Obrada izuzetaka

- Problem: okvir potomak će nasledeiti sve vrednosti roditelja ako se ove vrednosti namerno ne izmene.
- Na primer Jovan može da ima samo jednu nogu.
- Kada se projektuje sistem zasnovan na okvirima, svaki okvir koji je izuzetak od standardnog mora se eksplicitno obraditi.

# Višestruko nasleđivanje

- Na vrhu hijerarhijske strukture je jedna globalna klasa okvira koja opisuje opšti svet svih okvira, i obezbeđuje putem nasleđivanja informacije svim okvirima.
- U mnogim problemima je prirodno da se diskutuju objekti u pogledu veze sa raznim svetovima.
- Na primer, okvir Jovan može da se diskutovati kao deo sveta ljudskih bića i kao deo sveta zaposlenih u nekoj kompaniji.
- Objekat Jovan nasleđuje sada informacije od dva prethodnika ljudi i zaposlenih

# Višestruko nasleđivanje



# Facets

- Facet je prošireno znanje o svojstvu okvira.
- Na primer facets se mogu koristiti da uspostave početnu vrednost svojstva, da odrede njegov tip, ili ograniče moguće vrednosti.
- Mogu se koristiti da odrede načine na koji se dobija vrednost svojstva ili šta da se radi ako se ta vrednost promeni.



# Facets

- tag – definiše tip vrednosti koje se pridružuju svojstvu,
- implicitno – određuje implicitnu vrednost svojstva,
- dokumentacija – obezbeđuje dokumentaciju o svojstvu,
- ograničenje – određuje dozvoljene vrednosti,
- minimalna kardinalnost – određuje minimalna broj vrednosti koje neko svojstvo može da ima,
- maksimalna kardinalnost – određuje maksimalan broj vrednosti koje neko svojstvo može da ima,
- if-needed – specificira akciju koju treba preduzeti ako je potrebna vrednost svojstva,
- if-changed – izvršava neku metodu pri promenu vrednosti svojstva. Na primer, ako se vrednost svojstva Čitanje promeni, izvršava se metoda kojom se ažurira objekat displej koji predstavlja senzor1

# Facets - primer

ime objekta:

senzor1
---------

klasa:

temperaturni
--------------

svojstva:

čitanje	2
---------	---

aspekti:

minimalna kardinalnost	1
tip	numerički
if-needed	sakupljanje podataka
if-changed	Promena prikazivanja objekta

# Facets - primer

svojstvo:

lokacija	pumpa1
----------	--------

aspekti:

minimalna kardinalnost	1
maksimalna kardinalnost	1
ograničenja	pumpa1, pompa2, pompa3
if-needed	slike lokacije

časovnik za odmrzavanje

Status if-changed metod

status

On

if odmrzavanje.chasovnik.Status = On  
then hladnjak.Status = odmrzavanje

hladnjak

Status if-changed metod

status

odmrzavanje

if hladnjak.Status = odmrzavanje  
then kalem za zagrevanje.napajanje = On

kalem za

Napajanje if-changed metod

status

On

if kalem za zagrevanje.napajanje = On  
then kalem za zagrevanje.Status = On

# Metode

- Metoda je procedura koja je prododata objektu i biće izvršena kad god se zahteva.
- U opšetm slučaju if-needed metode se mogu formulisati tako da objekat dobija vrednost upitom korisnika, iz baze podataka, iz nekog algoritma, od drugog objekat, od drugog ekspertskog sistema.
- Metode se mogu napisati na nivou klase i naslediti od svih okvira na nižem nivou.

# Komunikacija između objekata

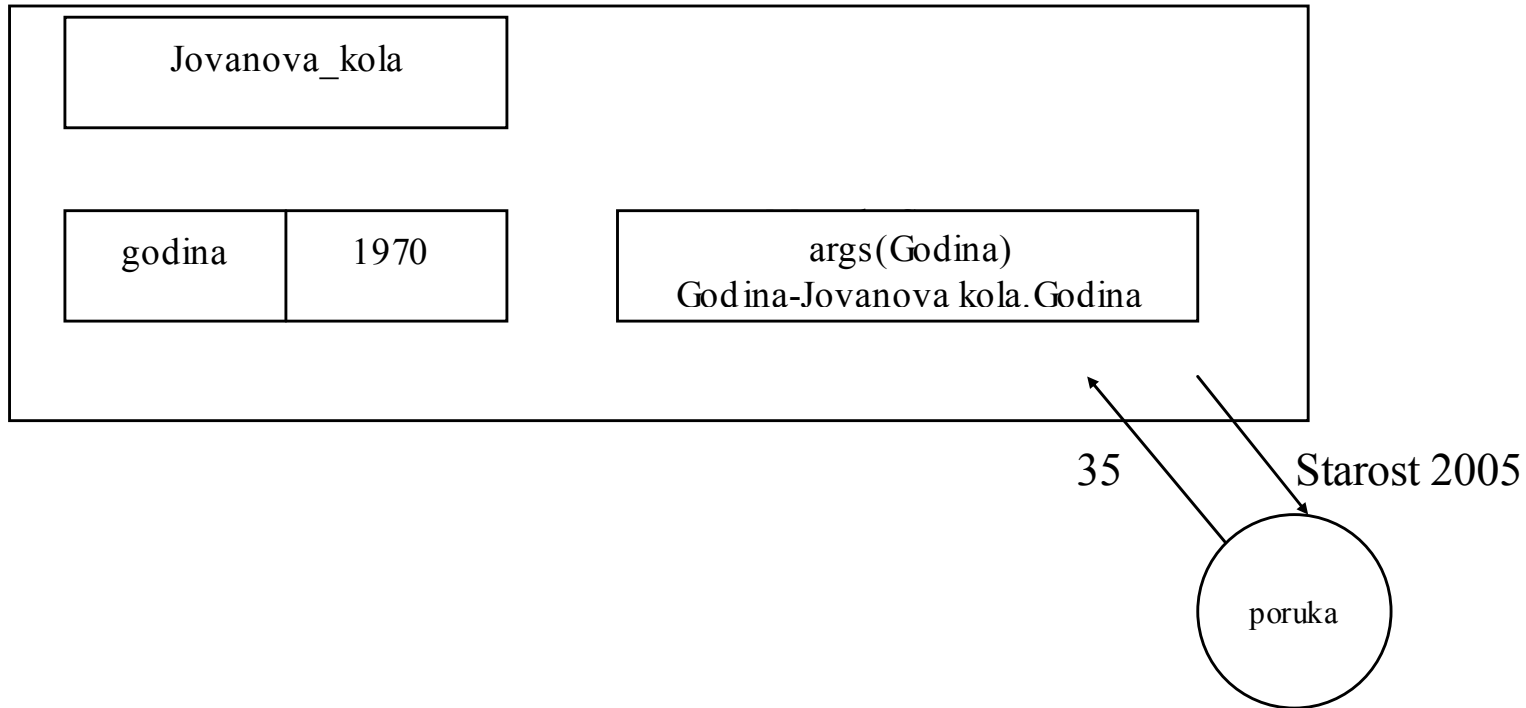
- Ako je nekom objektu potrebna informacija on je može dobiti od drugog objekta. Ili, ako se promeni vrednost atributa nekog objekta, to može da prouzrokuje promenu vrednosti atributa u nekom drugom objektu.
- Prethodnom šemom se pokazuje kako promena samo jedne vrednosti atributa dovodi do niza promena u brojnim objektima. Objekti mogu da utiču na vrednosti atributa u drugim objektima, pa i sopstvenih atributa

# Prosledjivanje poruka

- **Prosleđivanje poruka**
- Prosleđivanje poruka se sastoji u tome da se uputi signal objektu na koji objekat odgovara izvršavanjem metode, odnosno procedura.
- Poslati poruka *Crtaj* objektu *KRUG* - primi poruku, traži metodu *Crtaj* koja je za njega priključena, i eventualno odgovori crtanjem kruga.

send ime\_poruke ime\_objekta argumenti

# Prosledjivanje poruka



send Starost Jovanova\_kola 2005  
35



# Nasleđivanje ponašanja

- Mogu se naslediti i metode.
- Ako se te metode stave na nivo klase, onda se preko nasleđivanja dobija efikasan razvoj koda sistema.
- Nedostatak - metode se često pišu tako da rade sa podacima koji se nalaze u svojstvima okvira, pa se metod mora da se eksplicitno obrati imenu objekta i njegovom svojstvu - narušena opštost.

# Pisanje opštih metoda

- Na primer, okruženje će koristiti reč *Self* ili neku drugu da bi se vezalo ime primerka (instance) za metod:

Self svojstvo

args(godina): godina - Self.godina

- odnosno:

args(godina): godina - Jovanova\_kola.godina

# Primer nasleđivanja ponašanja

- Sistem koji je u stanju da nadgleda temperaturu procesa u raznim tačkama, uz alarmiranje, ako se nešto desi loše. Raspoloživi su temperaturni senzori kojima može pristupiti sistem za sakupljanje podataka. Ne zna se broj senzora
- Klasa temperaturnih senzora, pa instance  
Klasa senzora (svojstva):
  - ime - ime mesta koje se nadzire,
  - vrednost - čitanje temperature senzora,
  - kritična vrednost - maksimalno dozvoljena temperatura,
  - status - trenutno stanje (normalno, alarmno).

# Primer nasleđivanja ponašanja

temperaturni senzori

ime	?
vrednost	?
kritična vrednost	?
status	?

uzmi-vrednost(Self.ime)

vrednost if-needed metod

if Self.Status = Uzbuna  
then zvoni-Uzbuna

Status if-changed metod

if Self.vrednost > Self.kritična.vrednost  
then Self.Status = Uzbuna

vrednost if-changed metod

# Svojstva okvira kao podaci

- Klasa: if ljudi.noge = 2 then ljudi mogu da hodaju
- Primerak: if Jovan.noge = 2  
then Jovan može da hoda
- **Izmena vrednosti svojstava**
- if Jovan voli da pliva  
then Jovan.sport = plivanje

# Svojstva okvira kao podaci

- **Formiranje novog primerka**
- if Ana je ljuta na Jovana zato što ne voli jedrenje  
then MAKE okvir Dragan  
instance-of ljudi  
WITH sport = jedrenje
- **Uklanjanje okvira**  
if Ana je stvarno ljuta na Jovana  
then RETRACT okvir Jovan

# Korišćenje pravila sa promenljivama

Jovan

Ana

Dragan

noge	- 1
starost	35
pol	muški
adresa	Beograd
sport	plivanje
voli	?

	- 2
	30
	ženski
	Beograd
	jedrenje
	?

	- 3
	33
	muški
	Valjevo
	jedrenje
	?

# Korišćenje pravila sa promenljivama

- Okvir? X

instance-of ljudi

WITH adresa = Beograd

WITH starost = ?Starost

- Odgovor je:

okvir: Jovan

Ana

instance-of ljudi

instance-of ljudi

boravište = Beograd

Beograd

Starost = 35

30



# Korišćenje uparivanja u zaključcima

if Okvir? X

instance-of ljudi

WITH boravište = Beograd

WITH Starost = ?Starost

then Okvir? X

WITH Starost = ?Starost + 5

# Složeno uparivanje promenljivih

- Ako bilo koji čovek voli isti sport kao bilo koja žena, tada čovek voli ženu.

if Okvir? Osoba1

WITH pol = muški

WITH sport = ?sport1

AND

Okvir? Osoba2

WITH pol = ženski

WITH sport = ?sport1

then Okvir? Osoba1

WITH voli = ?Osoba2

# Složeno uparivanje promenljivih

Uparivanje 1

Okvir: Dragan(?Osoba1)  
pol = muški  
sport = jedrenje (?sport1)

Uparivanje 2

Okvir: Jovan (?Osoba1)  
pol = muški  
sport = plivanje (?sport1)

(Prva premisa)

Uparivanje 1

Okvir: Ana (?Osoba2)  
pol = ženski  
sport = jedrenje (?Osoba2)

(Druga premisa)

Ispaljivanje pravila prouzrokuje promene u okviru Dragan:

Okvir: Dragan (?Osoba1)  
voli = Ana (?Osoba2)

# Formiranje novih primeraka

Okvir: rezervacije

mesto = ?

ime = ?

if Okvir? Osoba1

    WITH sport = jedrenje

then

    MAKE okvir rezervacije

    WITH mesto = Herceg Novi

    WITH ime = Osoba1

# Formiranje novih primeraka

Okvir: rezervacije 1

Okvir:

rezervacije 2

mesto = Herceg Novi

mesto = Herceg

Novi

ime = Ana

ime = Dragan

# Interakcija pravila sa objektima

- Mnogi sistemi sa okvirima koriste niz pravila radi interakcije sa informacijama koje su sadržane u okvirima.
- Ovakav sistem se naziva hibridni, jer kombinuje okvire i pravila za predstavljanje znanja u domenu.
- Klauzule u pravilima mogu da sadrže promenljive koje se mogu iskoristiti za nalaženje uparujućih vrednosti među svim okvirima