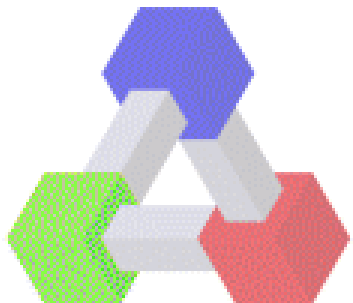


# Osnovi Računarske Inteligencije

---

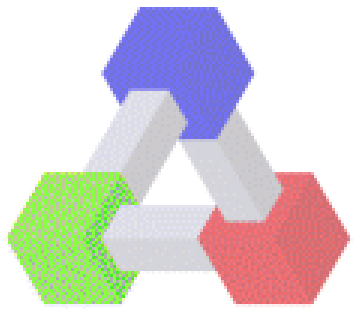
Predikatska logika  
PROLOG



# Sadržaj

---

- Proceduralno programiranje
- Deklarativno programiranje
- Reprezentacija znanja
- Zaključivanje
- Korišćenje



# Proceduralno programiranje

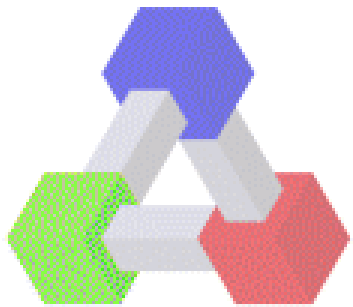
---

– Opisujemo način kako neki problem rešavamo

– Primer:

```
for(int i=0; i<20; i++) {  
    if(i%2==0)  
        printf("Broj %i je paran \n", i);  
    else  
        printf("Broj %i je neparan \n", i);  
}
```

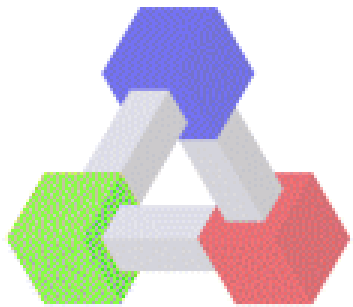
– Kako je ovde predstavljeno znanje?



# Deklarativno programiranje

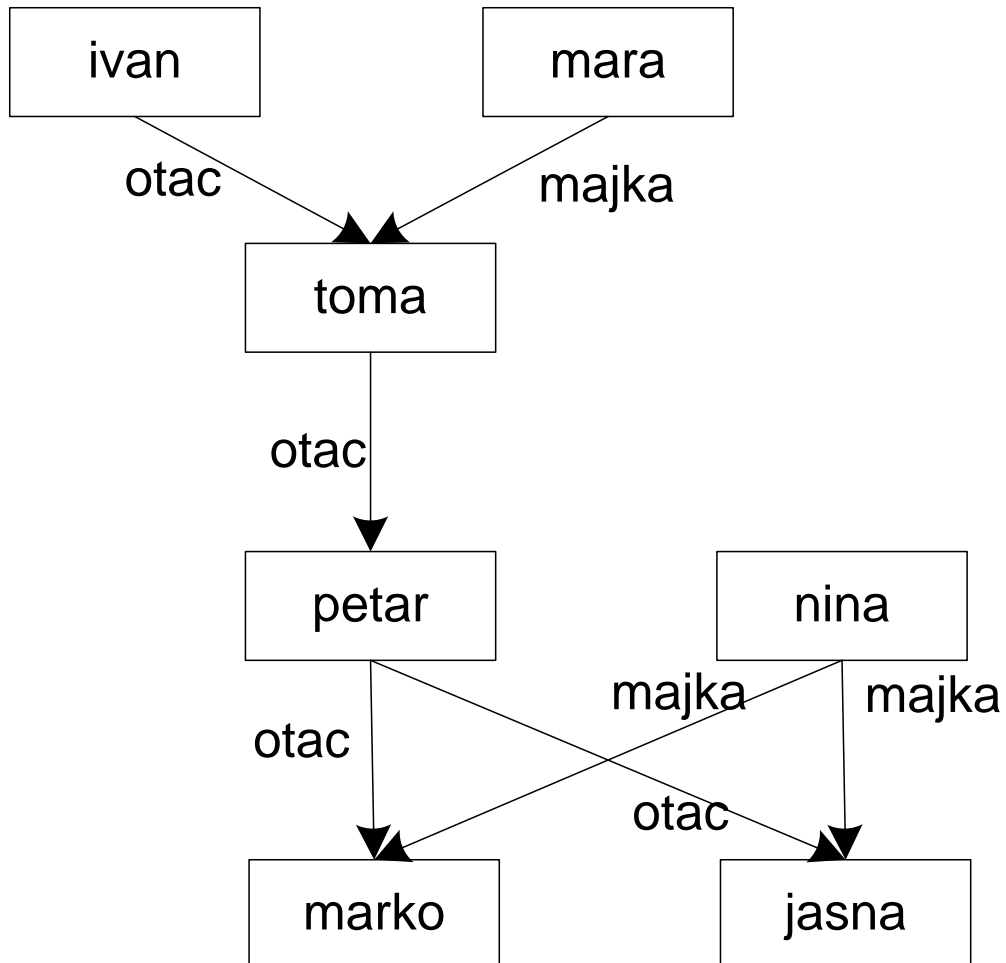
---

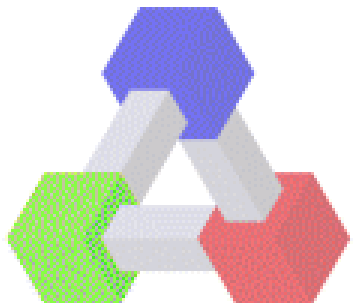
- Ivan je otac Tomi. Tomina majka je Mara. Tomino dete je Petar. Markov otac je Petar a Jasnina majka je Nina.
- Deda je očev ili majčin otac.
- Kome je Ivan deda?



# Deklarativno programiranje

---



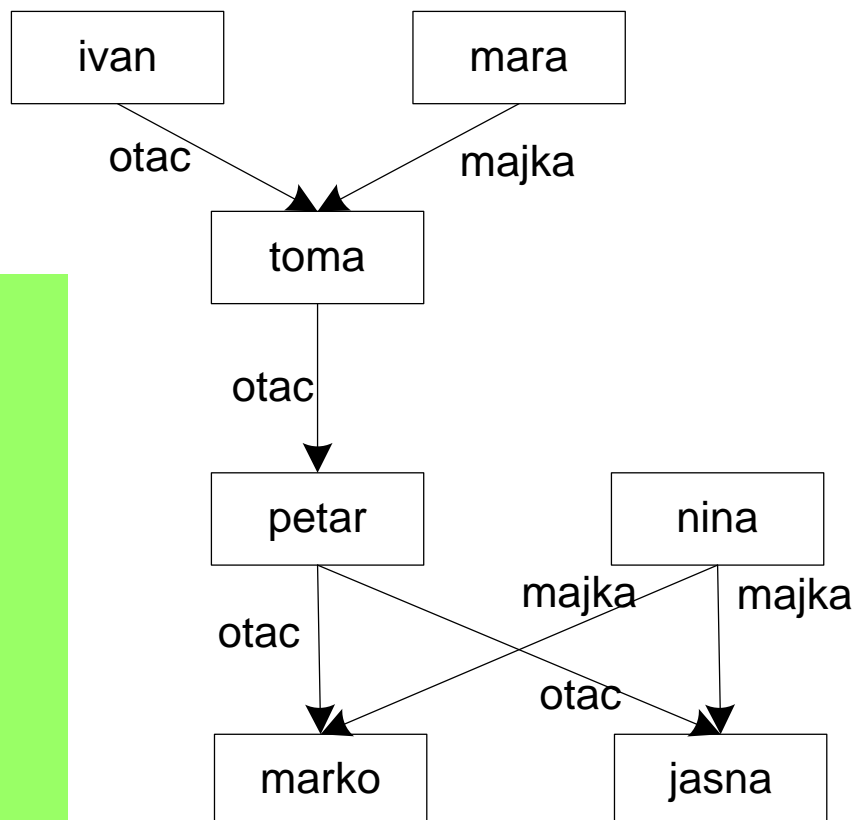


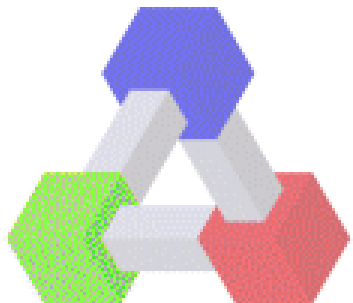
# PROLOG

## Baza znanja

Činjenice

```
otac(ivan, toma).  
otac(toma, petar).  
otac(petar, marko).  
otac(petar, jasna).  
majka(mara, toma).  
majka(nina, marko).  
majka(nina, jasna).
```



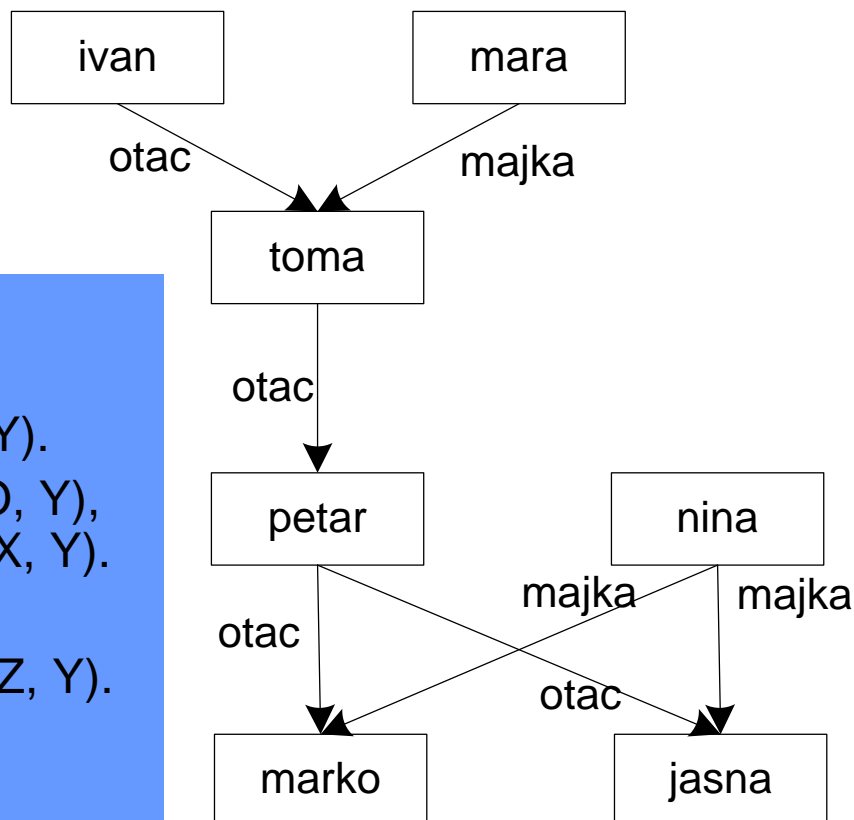


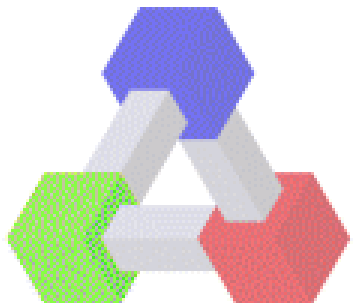
# PROLOG

## Baza znanja

### Pravila

```
roditelj(X, Y):- majka(X, Y).
roditelj(X, Y):- otac(X, Y).
deda(X, Y) :- otac(X, Z), roditelj(Z, Y).
sestra(X, Y):- roditelj(O, X), majka(O, Y),
               pol(Y, z), razliciti(X, Y).
predak(X, Y):- roditelj(X, Y).
predak(X, Y):- roditelj(X, Z), roditelj(Z, Y).
predak(X, Y):- roditelj(X,Z),
               roditelj(Z,W),
               roditelj(W,X).
```





# PROLOG

## Ciljevi ili upiti

Kome je Ivan deda?

Da li je Mara Petrova baka?

Ko je deda?

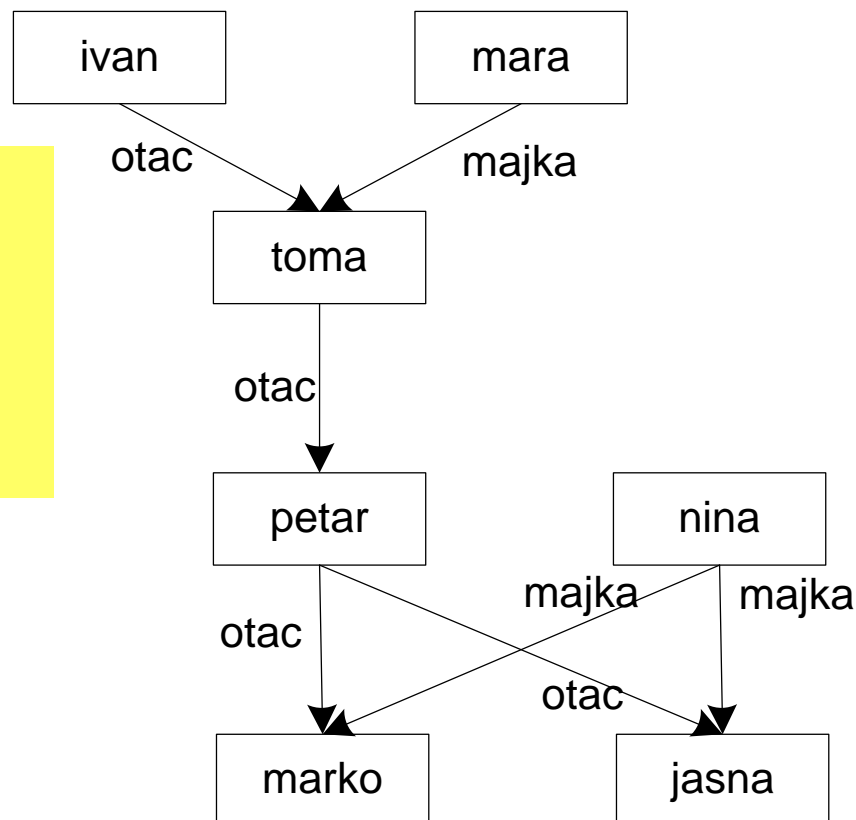
Koliko Toma ima godina?

?-deda(ivan, X).

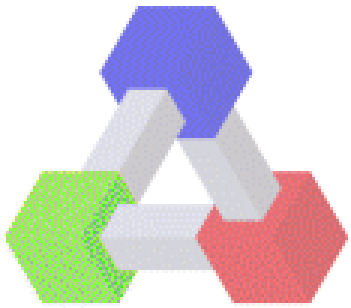
?-baka(mara, petar).

?-deda(X,\_).

?

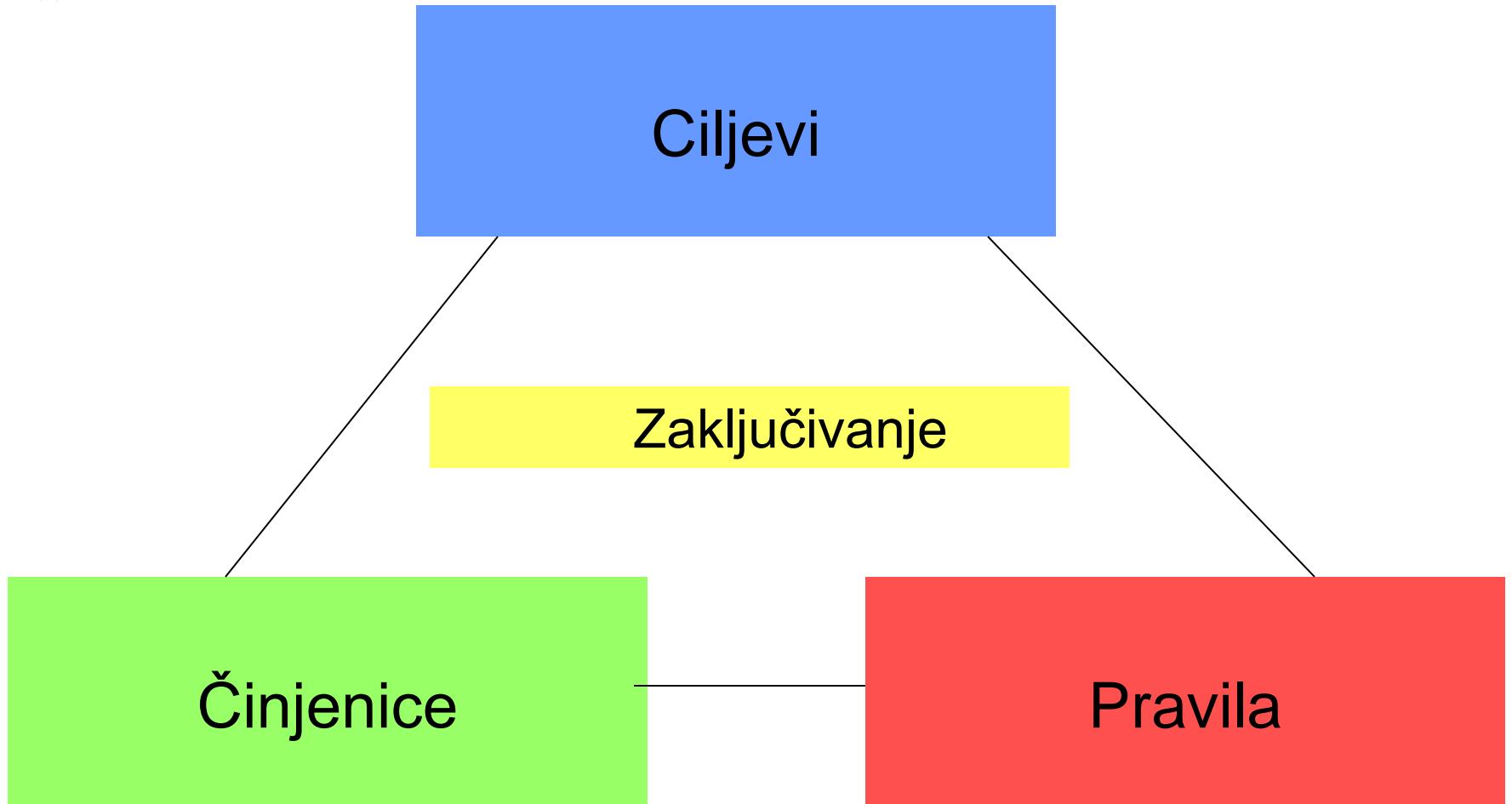


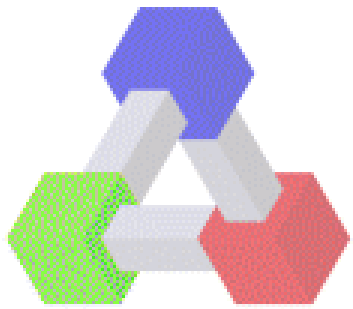




# PROLOG

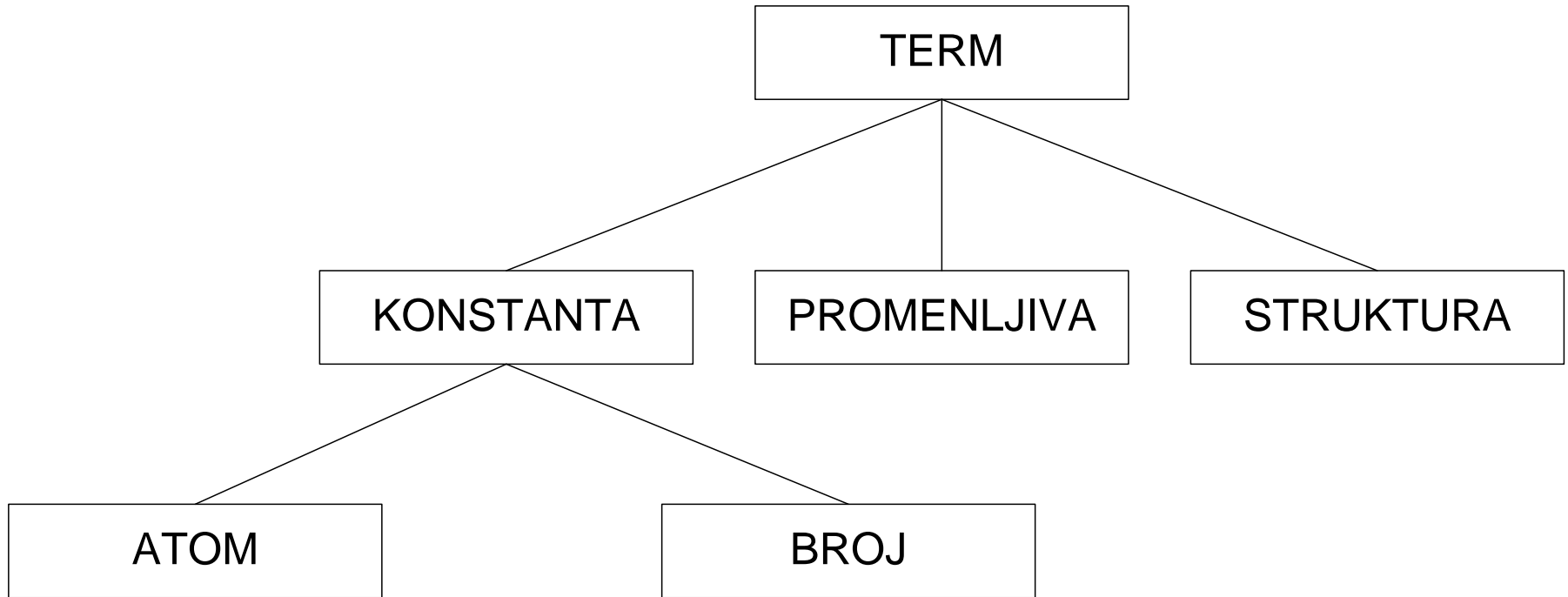
---

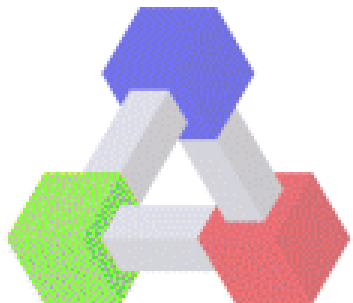




# Sintaksa programskog jezika

---

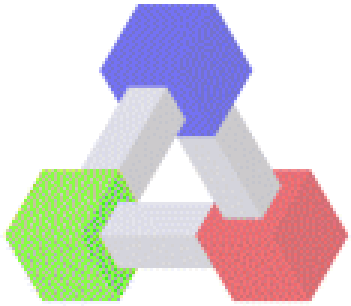




# Atom

---

*Atom* je svaki niz simbola, cifara i «\_» koji počinje malim slovom, ili je to niz posebnih znakova ili je to bilo koji niz znakova koji se nalazi unutar navodnika.

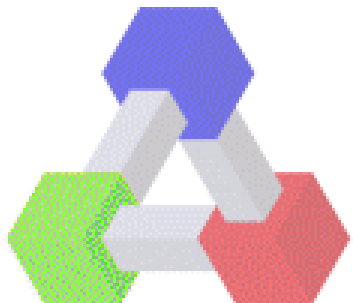


# Atom

---

- `pera`
- `automobil1`
- `'Automobil'`
- `avion_Boing`
- `avionBoing`

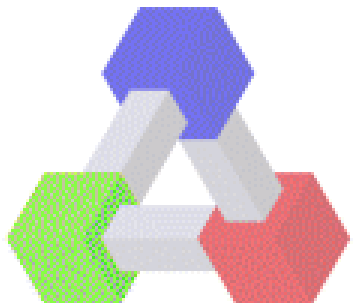
- `Pera`
- `avion boing`
- `3aviona`
- `_avion`
- `kanal(12,34)`



# Brojevi

---

*Brojevi* su niz cifara koji počinje predznakom i ne razlikuju se posebno celi brojevi od realnih.

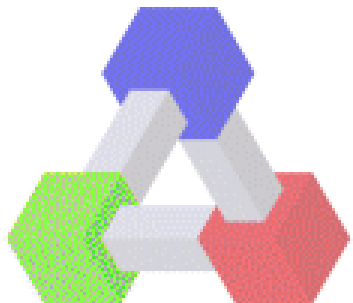


# Brojevi

---

- 32
- -24
- 13.007
- 23000.12
- -170001.45

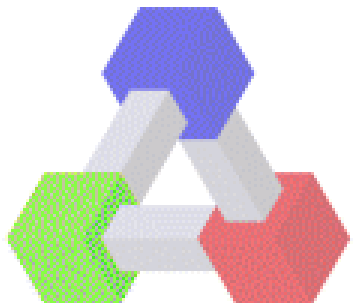
- 0xff1234
- 12,000.20
- $12+3i$
- $14 \cdot 10^6$
- jedan
- '12.3'



# Promenljiva

---

*Promenljive* su neke veličine koje se tokom izvršavanja menjaju. One se imenuju nizom znakova koji počinje velikim slovom ili specijalnim znakom `_`. Specijalan slučaj je `_` promenljiva koja predstavlja takozvanu anonimnu promenljivu, odnosno vrednost ove promenljive nam nije od interesa.



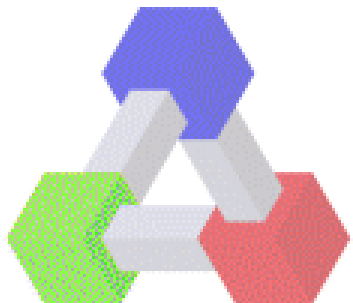
# Promenljiva

---

- X
- Y
- Otac
- \_promenljiva
- \_
- Z1a

- x
- promenljiva
- 'X'
- z1A

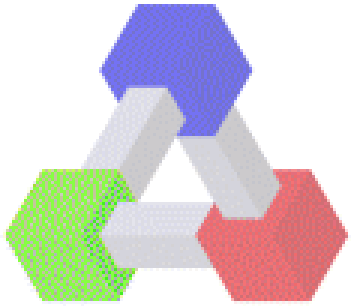




# Struktura

---

*Struktura* je složen term koji se formira vezivanjem nekoliko jednostavnijih termova. *Funktor* je atom koji objedinjuje strukturu u jedinstvenu celinu. Strukture se razlikuju po imenu i broju argumenata.

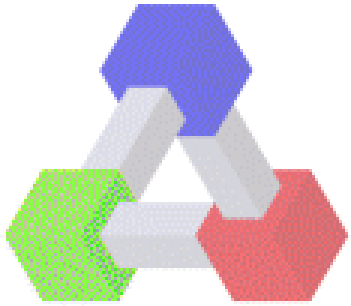


# Struktura

---

- `otac(pera, Mika)`
- `dan(23, 2, 2006)`
- `radnik(pera, status(bolovanje))`
- `sortiraj(Lista, Rez)`
- `or1A(1,at, x(1,s))`
- `q(X,Y):-r(X,Y), p(X)`
- `dan(dan(1,2), 3)`

- `1otac(pera, Mika)`
- `pera_otac_mika`
- `dan[23|2|2005]`
- `Sortiraj(lista,Rez)`
- `'otac(pera, Mika)'`



# Struktura

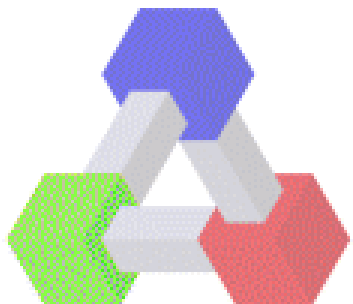
---

*specijalna\_veza(zaključak, pretpostavka)*

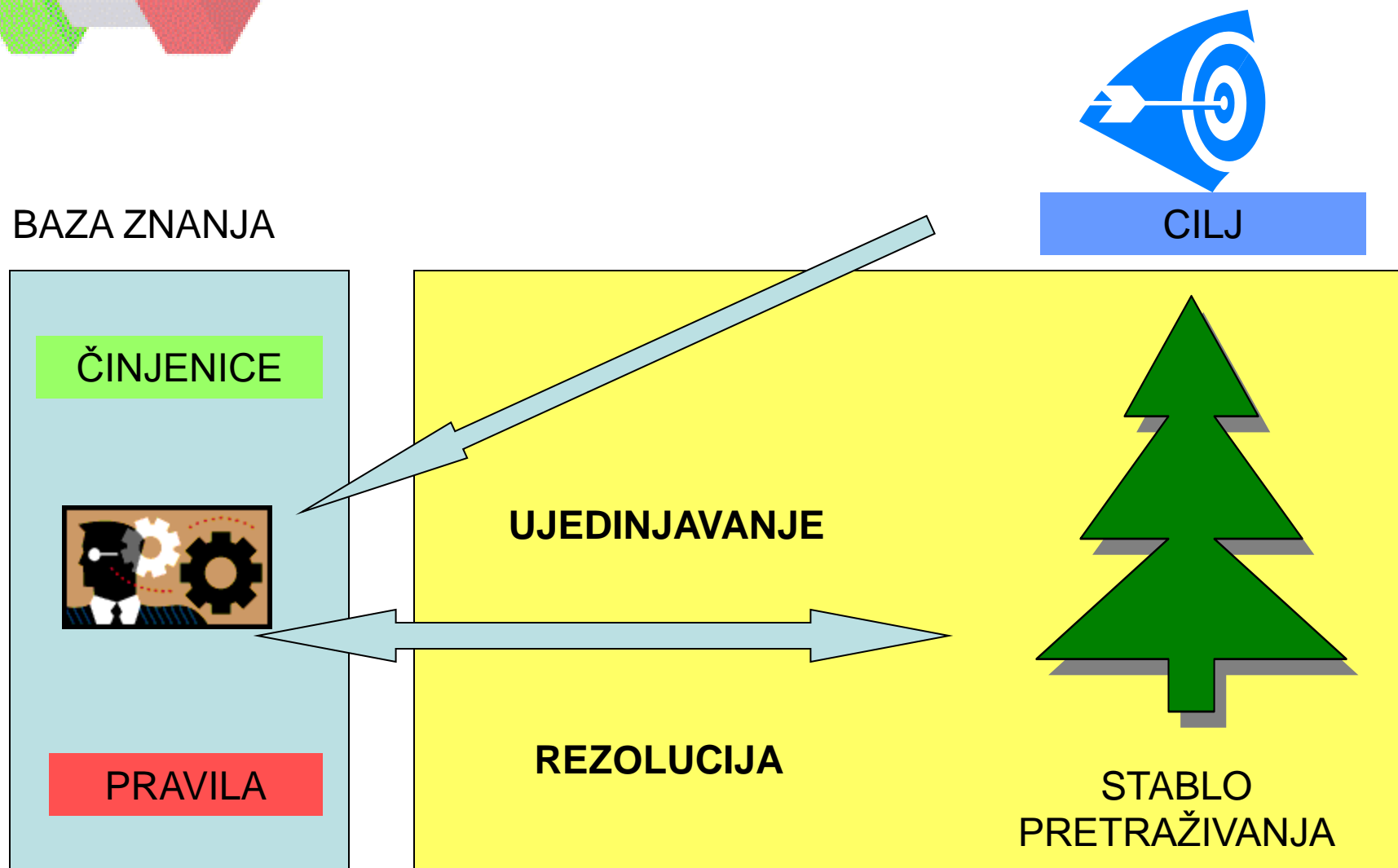
*:- (zaključak, pretpostavka)*

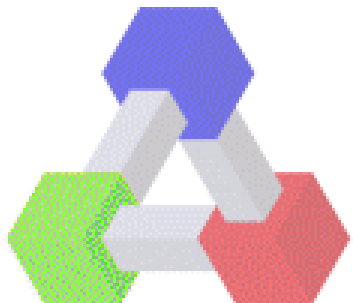
*zaključak :- pretpostavka*

*roditelj(X, Y):-otac(X, Y);majka(X, Y)*



# Izračunavanje odgovora





# Ujedinjavanje

## Supstitucija

- je konačan skup  $S$  parova oblika  $X = t$  za koje važi sledeće:
  - $\forall X$  se javlja samo u jednom paru iz  $S$
  - Promenljive nema sa desne strane znaka  $=$

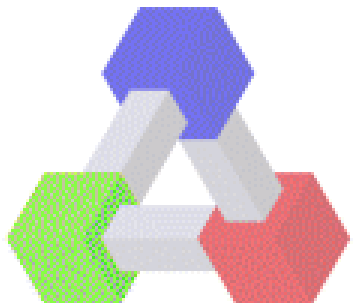
$A * S \quad A : \text{drug}(X, Y) .$

$S_1 = \{ X=\text{pera}, Y=\text{mika} \}$

$A * S_1 : \text{drug}(\text{pera}, \text{mika}) .$

$S_2 = \{ X=\text{brat}(\text{ana}), Y=\text{otac}(\text{ana}) \} \quad A * S_2 : \text{drug}(\text{brat}(\text{ana}), \text{otac}(\text{ana})) .$

$S_{2a} = \{ X=\text{pera}, X=\text{mika} \}$



# Ujedinjavanje

## Supstitucija

$$S_{2b} = \{ X=\text{pera}, Y=X \}$$

## Instanca

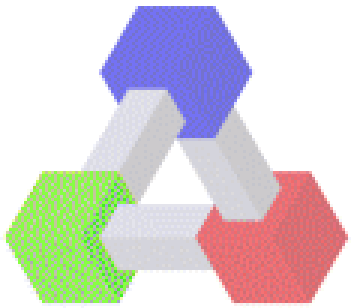
Term A je **instanca** terma B ako postoji supstitucija S za koju važi  $A=B*S$

$$B: p(X) :- q(X), r(X, Y) .$$

$$S_3 = \{ X=a, Y=b \}$$

$$A = B * S_3$$

$$A: p(a) :- q(a), r(a, b) .$$



# Ujedinjavanje

---

## Ujedinjivost

Dva su terma ujedinjiva ako imaju zajedičku instancu.

$$C = A * S = B * S$$

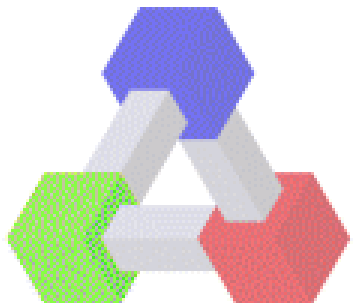
Primer

$A: p(a, Y) .$

$B: p(X, b) .$

$C = A * S_3 = B * S_3 ;$

$C: p(a, b) .$



# Ujedinjavanje

---

## Varijanta

Varijanta: Varijanta nekog terma  $A$  je instanca te strukture koja je dobijena primenom supstitucije koja sadrži samo promenljive.

### Primer

$p(X) : \neg q(X), \quad r(X) .$

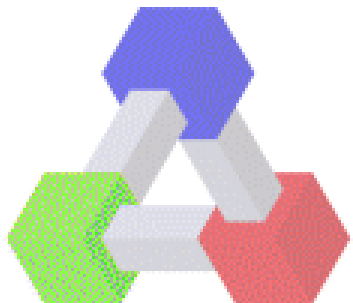
$P(Y) : \neg q(Y), \quad r(Y) .$

$p(K) : \neg q(K), \quad r(K) .$

$$S_6 = \{ X=Y \}$$

$$S_7 = \{ X=K \}$$



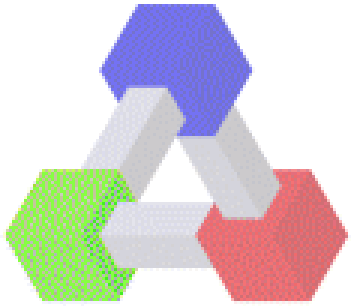


# Ujedinjavanje

---

## Proces ujedinjavanja dva terma

1. Dve strukture su ujedinjive ako imaju iste *funktore* i ako su svi argumenti jedne ujedinjive sa argumentima druge.
2. Ako su *termovi*  $A_1$  i  $A_2$  konstante onda su ujedinjivi samo ako su identični.
3. Ako je term  $A_1$  promenljiv a  $A_2$  proizvoljan term onda ujedinjavanje uspeva ako se promenljiva  $A_1$  ne pojavljuje u termu  $A_2$ . Pri ujedinjavanju  $A_1$  sa  $A_2$  promenljiva  $A_1$  poprima vrednost terma  $A_2$ . Ako je obrnuto isto važi.



# Ujedinjavanje

---

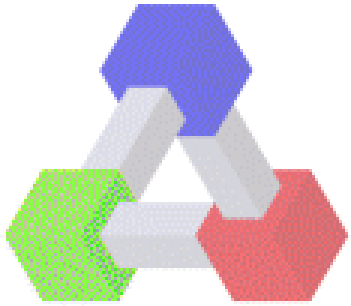
## Proces ujedinjavanja dva terma

drug(a, b).  
drugar(a, b).

drug(a, b).  
drug(a, b, c).

brz(X).  
brz(pas(X)).

brz(X).  
brz(pas(Y)).



# Ujedinjavanje

## Mehanizam redukovanja cilja

Procedura rezolviranja se svodi na pojednostavljivanje zadatog cilja korišćenjem ujedinjavanja sve dok ne dobijemo prazan cilj.

(k1)  $p(X, Y) :- q(X), r(X, Y)$ .

(k2)  $p(d, 4)$ .

(k3)  $q(a)$ .

(k4)  $q(b)$ .

(k5)  $q(c)$ .

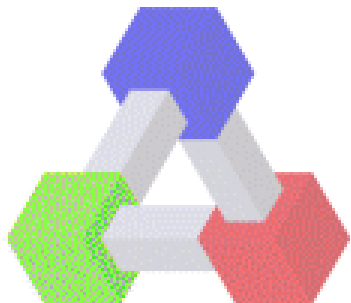
(k6)  $r(a, 1)$ .

(k7)  $r(a, 2)$ .

(k8)  $r(c, 3)$ .

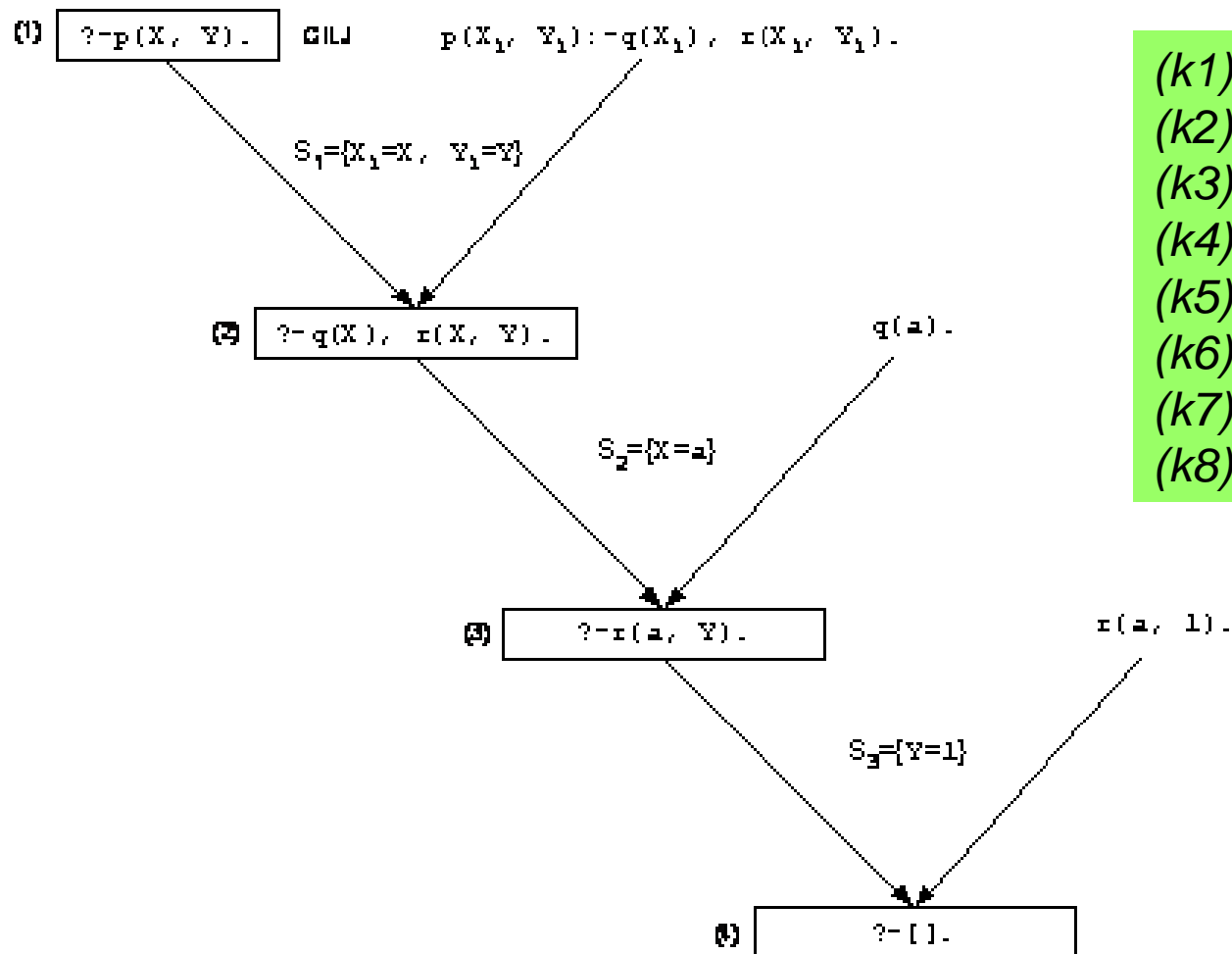
?- $p(X, Y)$

Kako da dobijemo koliko je X i Y?

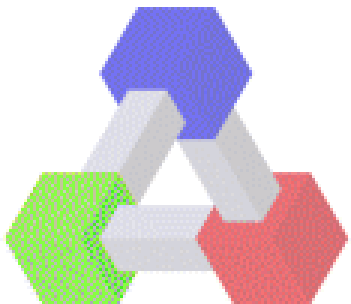


# Ujedinjavanje

## Mehanizam redukovanja cilja



(k1)  $p(X, Y) :- q(X), r(X, Y).$   
(k2)  $p(d, 4).$   
(k3)  $q(a).$   
(k4)  $q(b).$   
(k5)  $q(c).$   
(k6)  $r(a, 1).$   
(k7)  $r(a, 2).$   
(k8)  $r(c, 3).$



# Stablo pretraživanja

1. *Koren stabla* je polazni cilj  $C$ .

2. Neka je cilj oblika  $?-C1, C2, \dots, Cn$ . čvor stabla pretraživanja i neka je  $C1$  pozitivan pod cilj tada taj čvor ima po jednog potomka za svaku klauzulu oblika  $G:-T1, \dots, Tq$ . za koju su strukture  $C1$  i  $G$  ujedinjive.

Tada potomak cilja  $?-C1, C2, \dots, Cn$ . izgleda

$?-(T1, \dots, Tq, C2, \dots, Cn)*S$

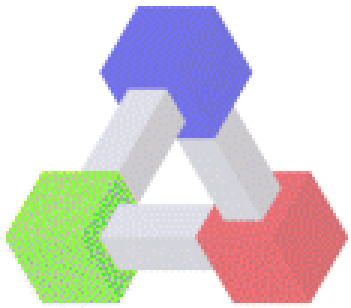
Gde je  $S$  najopštiji ujedinitelj struktura  $C1$  i  $G$ .

Ujedinjive klauzule traže se odozgo prema dolja i s leva na desno.

3. Neka je cilj  $?-C1, C2, \dots, Cn$ . čvor stabla pretraživanja ali je  $C1$  oblika  $C1 = \text{not}(C1^*)$ . Ako cilj ima  $C1^*$  konačno stablo pretraživanja sa granom uspeha tada celi čvor nema potomaka.

Međutim, ako  $C1^*$  ima konačno stablo pretraživanja ali bez grane uspeha onda je potomak cilj  $?-C2, \dots, Cn$

4. Potomak koji nema ni jednog elementa naziva se prazan cilj  $[]$



# Stablo pretraživanja

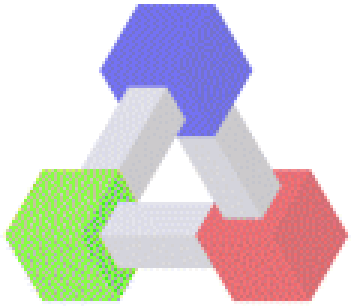
---

**Čvor uspeha** je čvor koji sadrži prazan cilj.

Grana stabla koje završavaju na čvoru uspeha nazivaju se **grane uspeha**.

Čvor koji sadrži ne prazan cilj a nema potomaka naziva se **čvor neuspeha**.

**Grana neuspeha** je grana koja završava na čvoru neuspeha.



# Stablo pretraživanja

---

(k1)  $p(X, Y):-q(X), r(X, Y).$

(k2)  $p(d, 4).$

(k3)  $q(a).$

(k4)  $q(b).$

(k5)  $q(c).$

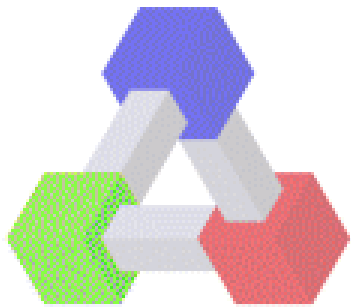
(k6)  $r(a, 1).$

(k7)  $r(a, 2).$

(k8)  $r(c, 3).$

?- $p(X, Y)$

Kako da dobijemo koliko je X i Y?



# Stablo pretraživanja

(k1)  $p(X, Y) :- q(X), r(X, Y).$

(k2)  $p(d, 4).$

(k3)  $q(a).$

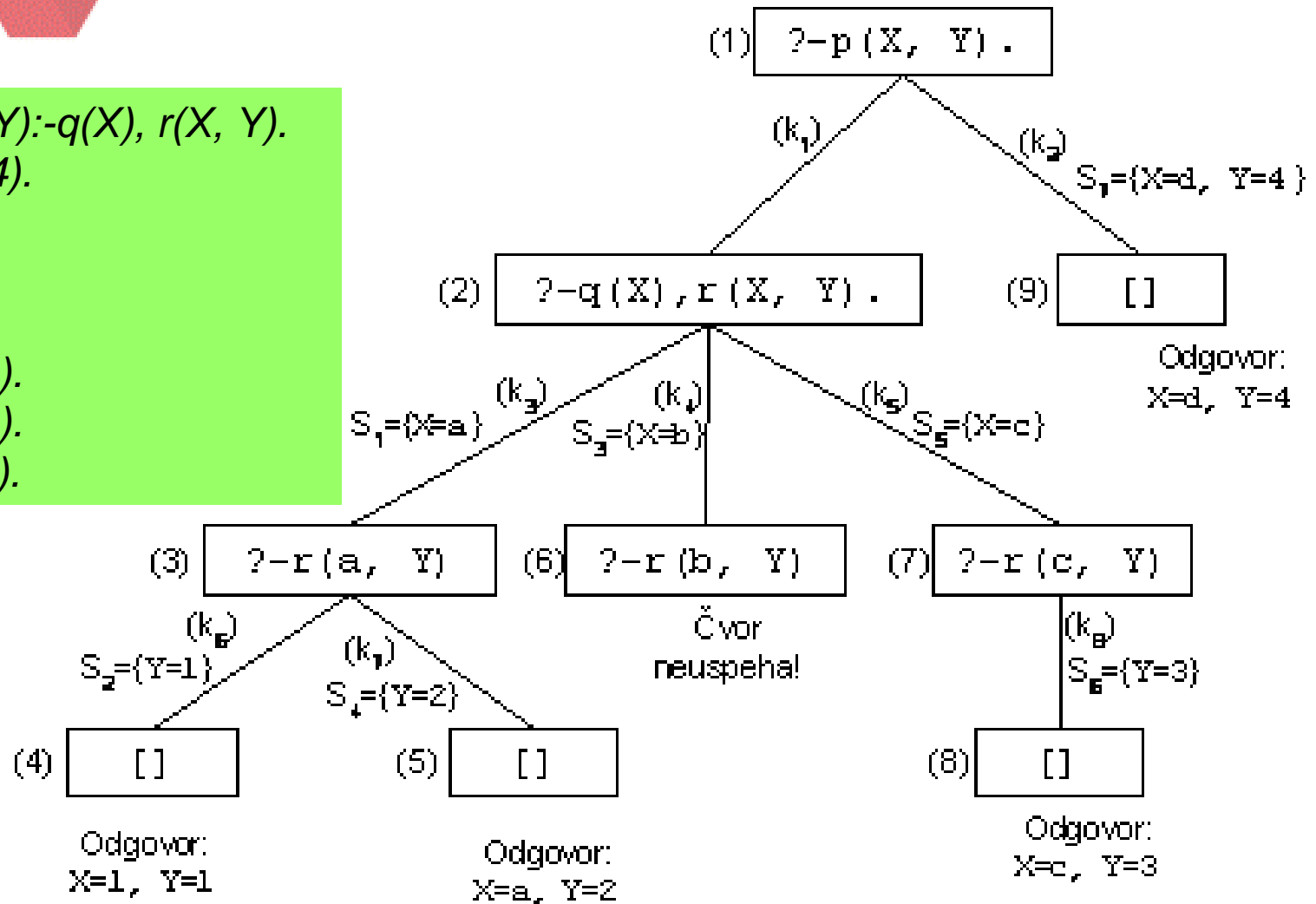
(k4)  $q(b).$

(k5)  $q(c).$

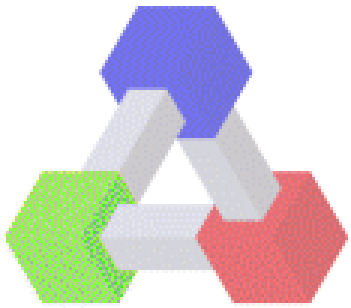
(k6)  $r(a, 1).$

(k7)  $r(a, 2).$

(k8)  $r(c, 3).$



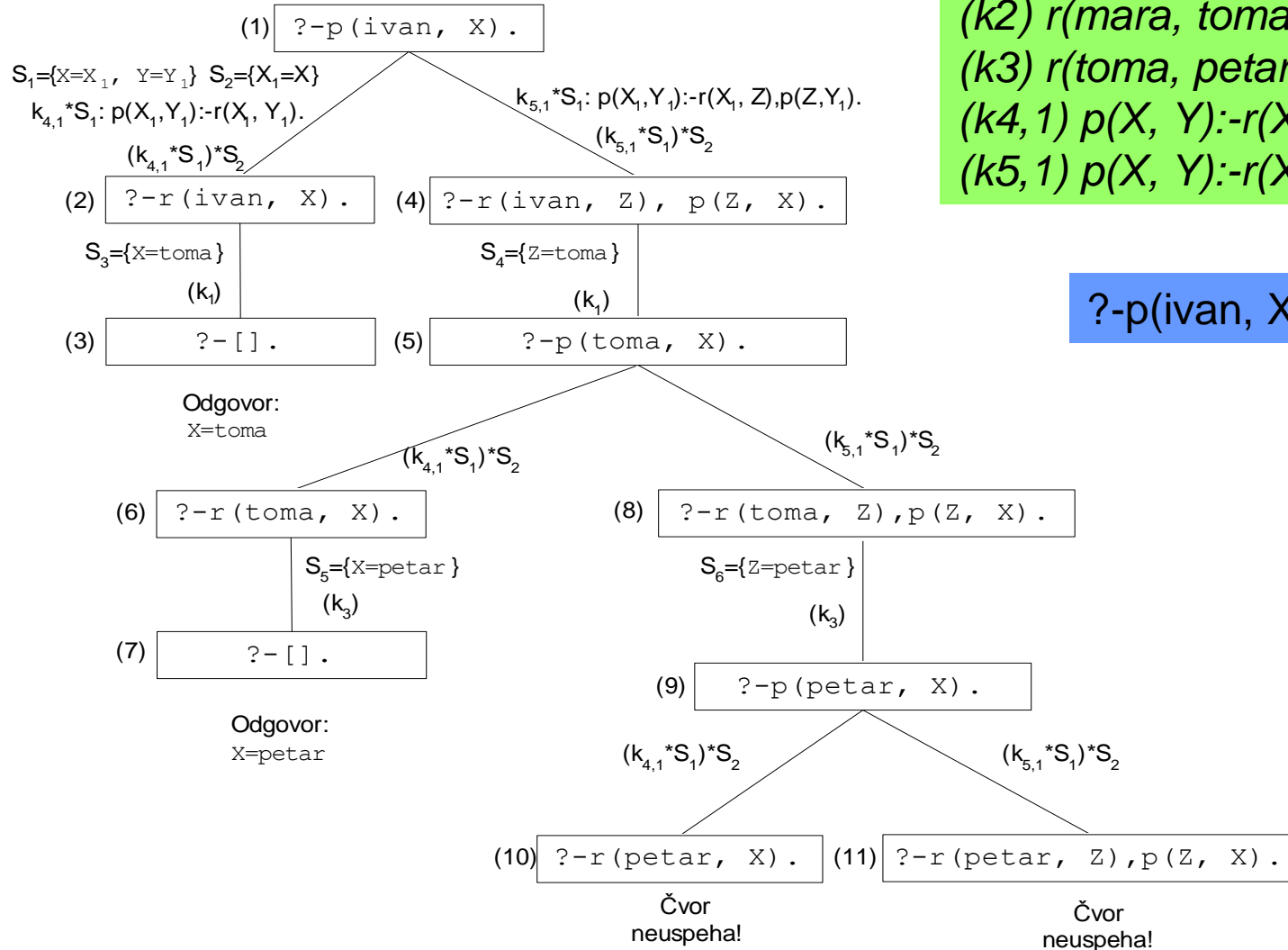


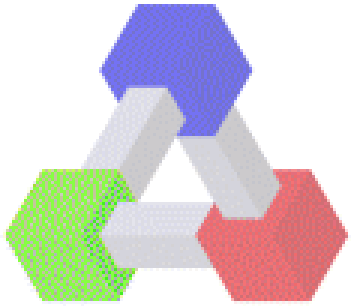


# Stablo pretraživanja

$(k1) r(ivan, toma).$   
 $(k2) r(mara, toma).$   
 $(k3) r(toma, petar).$   
 $(k4, 1) p(X, Y):-r(X, Y).$   
 $(k5, 1) p(X, Y):-r(X, Z), p(Z, Y).$

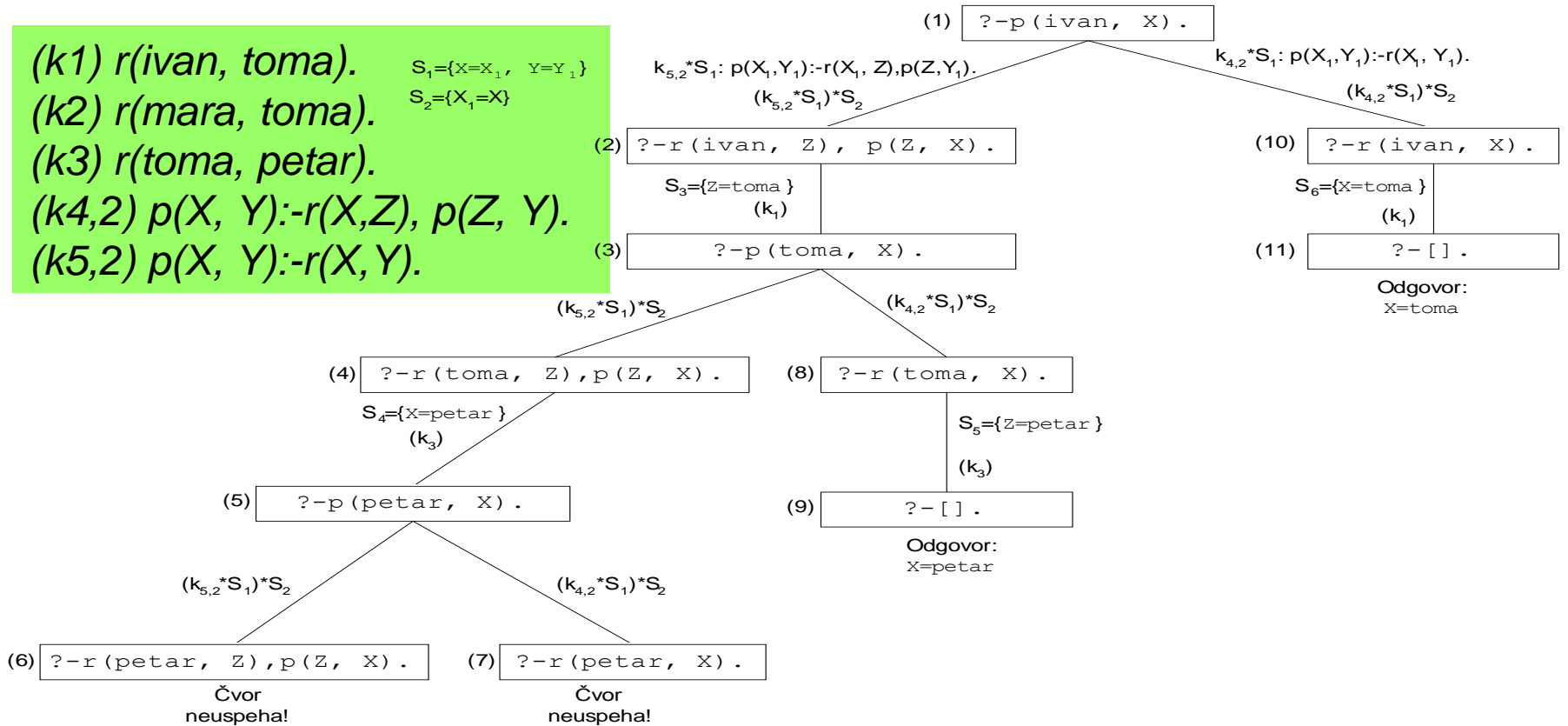
$?-p(ivan, X).$



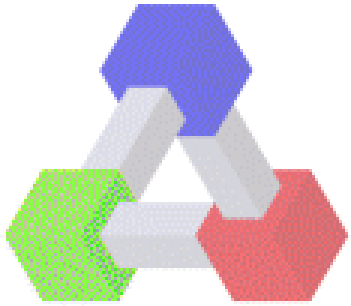


# Stablo pretraživanja

$(k1) r(ivan, toma).$   $S_1=\{X=X_1, Y=Y_1\}$   
 $(k2) r(mara, toma).$   $S_2=\{X_1=X\}$   
 $(k3) r(toma, petar).$   
 $(k4,2) p(X, Y):-r(X,Z), p(Z, Y).$   
 $(k5,2) p(X, Y):-r(X,Y).$

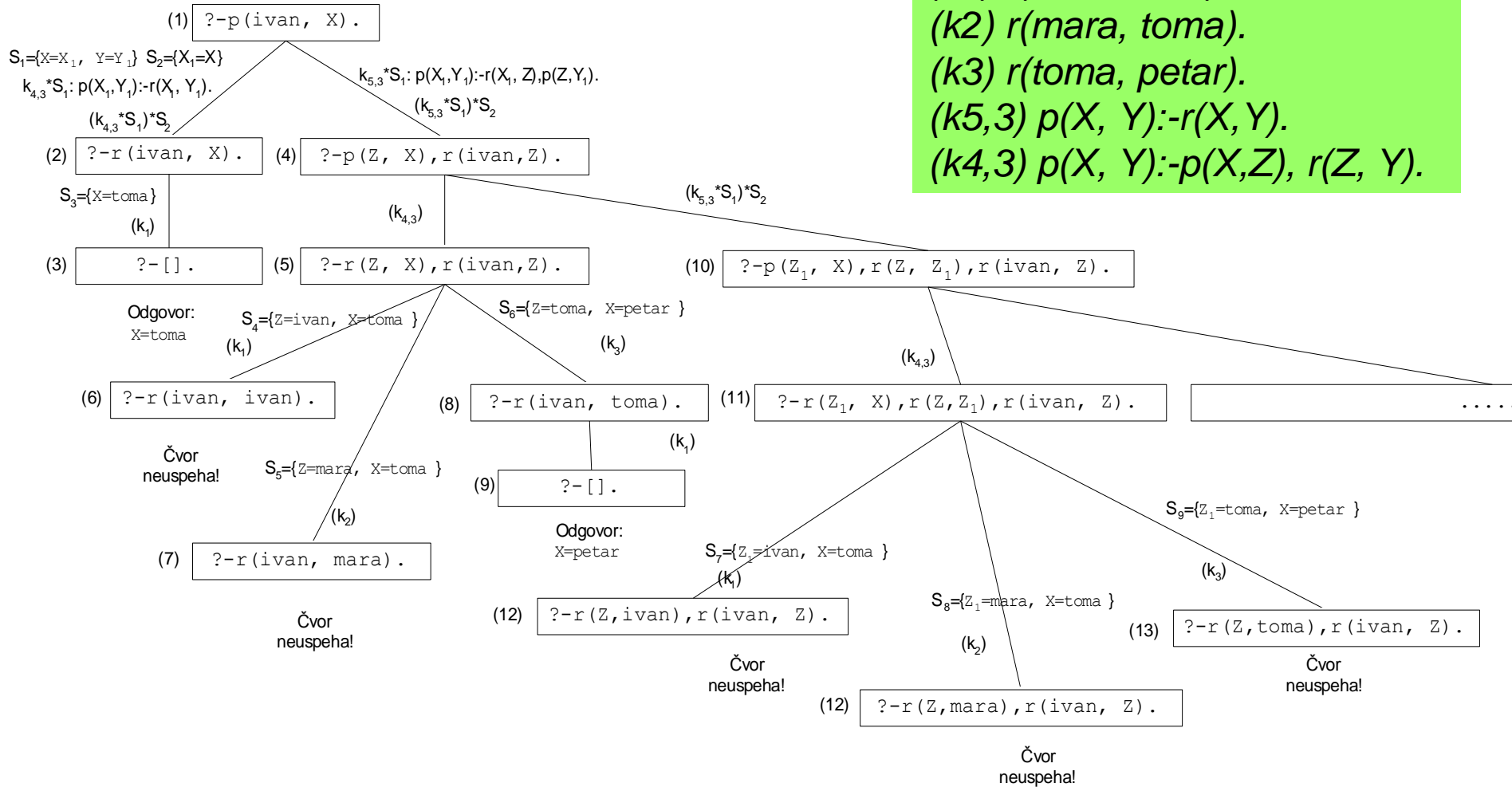


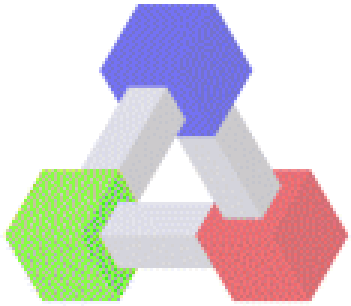
$?-predak(ivan, X).$



# Stablo pretraživanja

$(k1) r(ivan, toma).$   
 $(k2) r(mara, toma).$   
 $(k3) r(toma, petar).$   
 $(k5,3) p(X, Y):-r(X, Y).$   
 $(k4,3) p(X, Y):-p(X,Z), r(Z, Y).$





# Liste

---

a, b, c, d

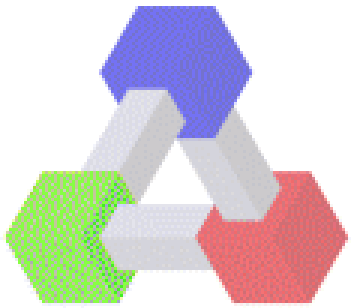
`st(a, st(b, st(c, p)))`

`[a | [b | [c | []]]]`

`[a,b,c]`

`[Glava | Rep]`

`[]` *prazna lista, lista bez elemenata*



# Rad sa listama

---

Provera da li je struktura lista

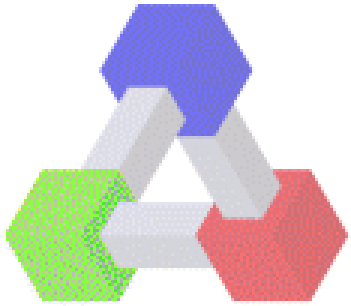
```
lista([ ]).  
lista([G|R]):-lista(R).
```

Traženje elementa u listi

```
element(X, [X|_ ]).  
element(X, [Y|R]):-element(X,R).
```

Spajanje dve liste

```
pripoji([], X, X).  
pripoji([X|R], Y, [X|Z]):-pripoji(R, Y, Z).
```



# Rad sa listama

---

## Primeri korišćenja

```
?-lista([ ]).
```

```
yes
```

```
?-lista([a, r, 1, 2, dan(23,4)]).
```

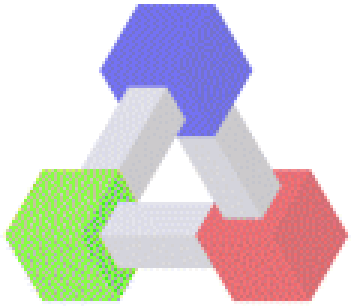
```
yes
```

```
?-element(b, [a, b, c]).
```

```
yes
```

```
?-element(X, [a, b, c]).
```

```
X=a; X=b; X=c
```



# Rad sa listama

---

Primeri korišćenja

```
pripoji([], X, X).
```

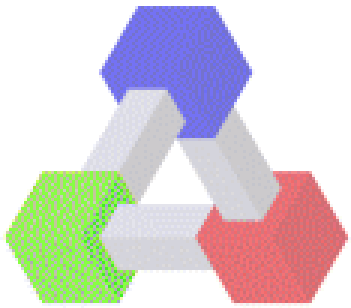
```
pripoji([X|R], Y, [X|Z]):-pripoji(R, Y, Z).
```

```
?-pripoji([a, b, c], [d,e], X).
```

```
X=[a,b,c,d,e]
```

```
?-pripoji([a,c], X, [a,c,d,e,f]).
```

```
X=[d,e,f]
```



# Rad sa listama

---

## Obrtanje elemenata liste

```
obrni([], []).  
obrni([X|Xr], Y):-obrni(Xr, Yr), pripoji(Yr, [X], Y).  
obrni1(L, OL):-obrni(L, [], OL).  
obrni([], OL, OL).  
obrni([G|R], S, OL):-obrni(R, [G|S], OL).
```

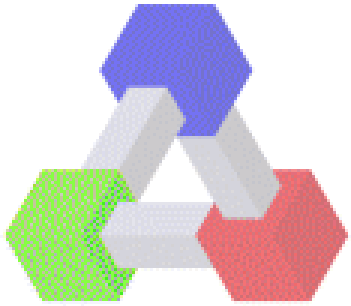
## Brisanje elementa iz liste

```
brisi(X, [X|R], R).  
brisi(X, [Y|R], [Y|R1]):-brisi(X, R, R1).
```

## Dodavanje elementa u listu

```
dodaj(X, L, VL):-brisi(X, VL, L).
```





# Rad sa listama

---

## Permutacija elemenata liste

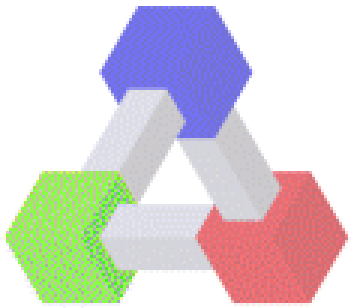
```
permutacija([], []).  
permutacija([X|L], P):-permutacija(L, L1), ubaci(X, L1, P).
```

## Provera da li je lista uređena

```
uredjena([X]).  
uredjena([X|[Y|R]]):-X=<Y, uredjena([Y|R]).
```

## Sortiranje

```
sortiraj(L, SL):-permutacija(L, SL), uredjena(SL).
```



# Rad sa listama

---

## Podlista

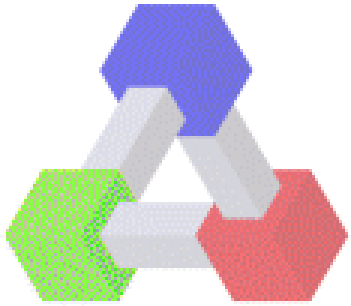
```
podlista(S, L):-pripoji(L1, S, L2), pripoji(L2, L3, L).
```

## Broj elemenata liste

```
broj_elementata([], 0).  
broj_elementata([G|R], N):- broj_elementata(R, N1), N is N1 + 1.
```

## Suma elemenata liste

```
suma_elementata([], 0).  
suma_elementata([G|R], N):- suma_elementata(R, N1), N is N1 + G.
```

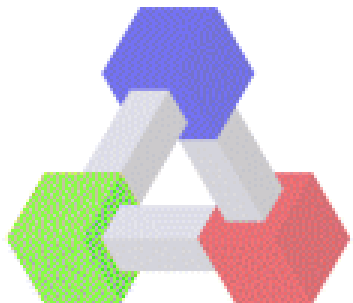


# Rad sa listama

---

Quick sort algoritam

```
qsort([], []).  
qsort([G|R], T):-manji(G, R, LM),  
    veci(G, R, LV),  
    qsort(LM, SLM),  
    qsort(LV, SLV),  
    pripoji(SLM, [G|SLV], T).  
manji(G, [], []).  
manji(X, [G|R], [G|R1]):-G=<X, manji(X, R, R1).  
manji(X, [G|R], R1):-G>X, manji(X, R, R1).  
veci(G, [], []).  
veci(X, [G|R], [G|R1]):-G>=X, veci(X, R, R1).  
veci(X, [G|R], R1):-G<X, veci(X, R, R1).
```



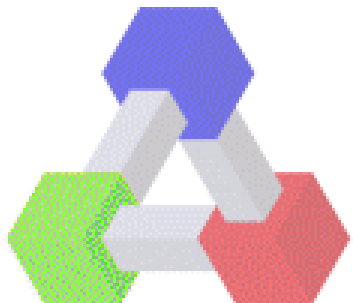
## Diferenciranje matematičkih izraza

---

$$f(x) = (2 * x + 1)^5 + (x + 1)^4$$

$$\frac{df(x)}{dx} = ?$$

$$5 * (2 * x + 1)^4 * 2 + 4 * (x + 1)^3$$



# Diferenciranje matematičkih izraza

---

## Tablica osnovnih izvoda

izvod(x, 1).

izvod(x^N, N\*x^N1):-integer(N), N>1, N1 is N-1.

izvod(N, 0):-integer(N).

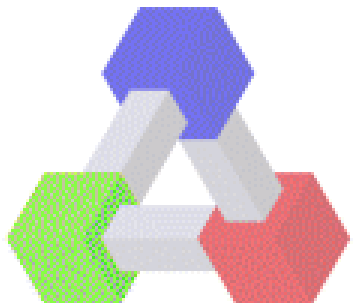
izvod(U^N, N\*U^N1\*Du):-integer(N), not(N=0), N1 is N-1, izvod(U, Du).

izvod(U+V, Du+Dv):-izvod(U, Du), izvod(V, Dv).

izvod(U\*V, Du\*V+U\*Dv):-izvod(U, Du), izvod(V, Dv).

?-izvod((2\*x+1)^5, T).

T = 5\* (2\*x+1)^4\* (0\*x+2\*1+0)



# Diferenciranje matematičkih izraza

---

## Tablica osnovnih izvoda

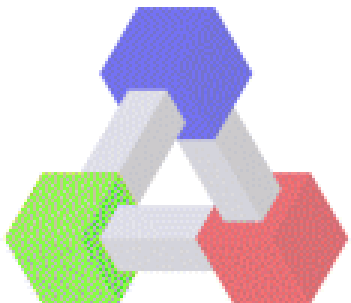
izvod( $\sin(U)$ ,  $Du \cdot \cos(U)$ ):-izvod( $U$ ,  $Du$ ).

izvod( $\cos(U)$ ,  $-Du \cdot \sin(U)$ ):-izvod( $U$ ,  $Du$ ).

?- izvod( $\sin(x)^3$ ,  $T$ )

$T = 3 \cdot \sin(x)^2 \cdot (1 \cdot \cos(x))$  .

integral( $U$ ,  $V$ ):-izvod( $V$ ,  $U$ ).



# Ciljevi u prologu

---

Ako je dat program:

```
otac(mirko, vladimir).  
otac(vladimir, petar).  
otac(petar, milan).  
majka(milena, vladimir).  
majka(sonja, petar).  
majka(marija, milan).  
roditelj(X,Y):-majka(X, Y).  
roditelj(X,Y):-otac(X,Y).  
predak(X, Y):-roditelj(X, Y).  
predak(X, Y):-roditelj(X, Z),  
                predak(Z, Y).
```

Kako biste u PROLOG-u postavili pitanje:

1. Ko su očevi?
2. Da li je milena majka?
3. Kome je Mirko otac?
4. Ko je Milanov otac?
5. Da li je Sonja Petrova majka?
6. Ko su Milanovi preci?
7. Kome je Vladimir roditelj?
8. Ko je kome majka?
9. Ko je Petrov roditelj?