

Pristup nelokalnim imenima

- **Pravila djelokruga**
 - određuju važeću deklaraciju **variabile, procedure, polja, ili nekog drugog identifikatora koji nije lokalno deklariran**
- **Pravila:**
 - **Statički djelokrug bez ugniježđenih procedura**
 - **Statički djelokrug ugniježđenih procedura**
 - **Dinamički djelokrug**

Pristup nelokalnim imenima

- **Statičko pravilo djelokruga bez ugniježđenih procedura**
 - Sve procedure deklarirane su isključivo naredbama glavnog programa
 - **Primjer**
 - Programski jezik C
 - Napomena
 - sintaksa jezika C zahtijeva da su sve procedure uključujući i glavni program nezavisno deklarirane
 - deklaracije procedura nisu statički ugniježđene u glavnom programu
 - **Identifikator nije deklariran naredbama procedure**
 - potrebno ga je deklarirati naredbama glavnog programa
 - **Pravilo:**

**Ako ime nije deklarirano naredbama procedure,
onda je važeća deklaracija zadana naredbama glavnog programa**

- **Statičko**
 - djelokrug deklaracije određuje se statičkom analizom tijekom prevođenja izvornog u ciljni program

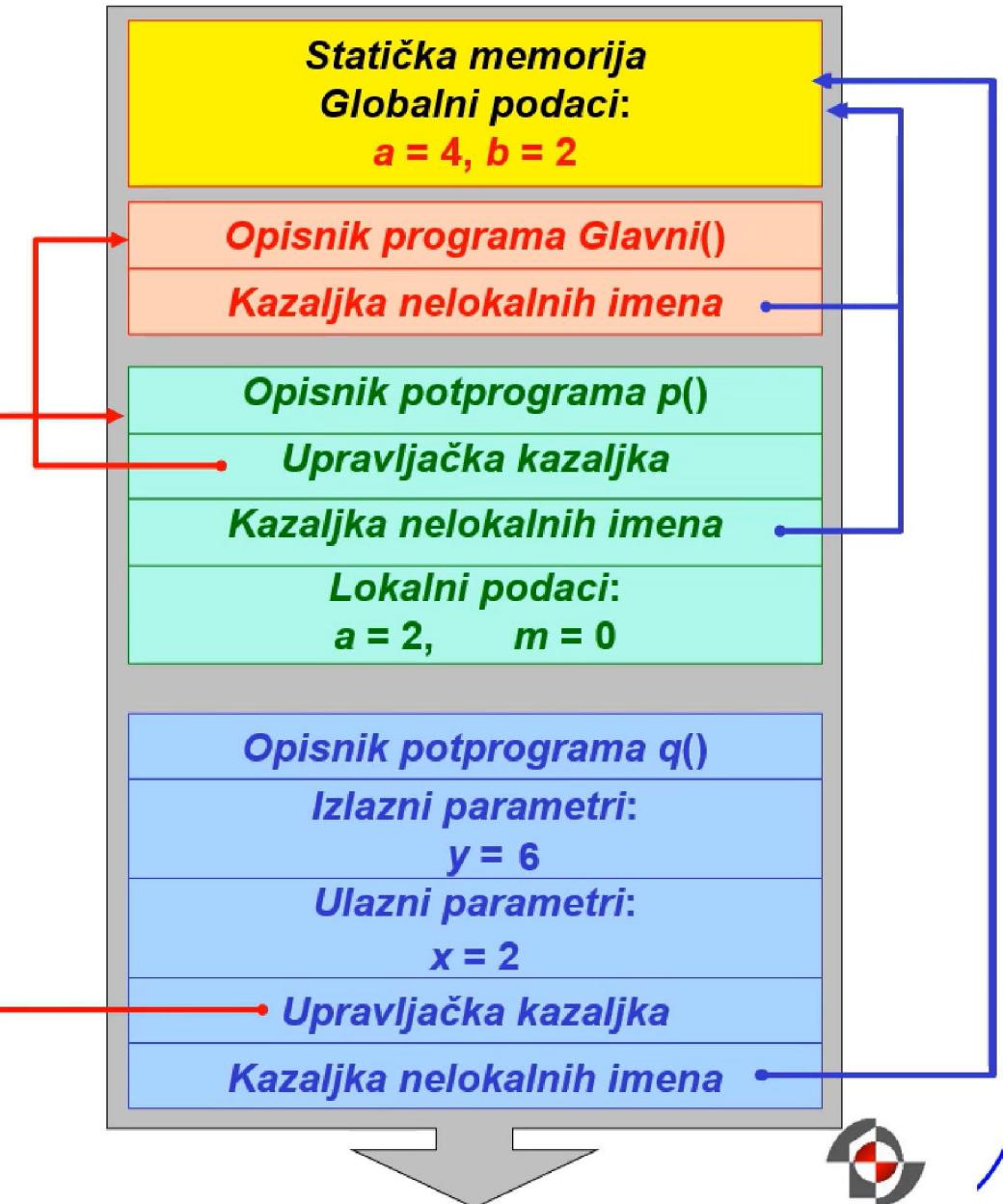
Statičko pravilo djelokruga bez ugniježđenih procedura

```
(1) Glavni()
(2) int a=4, b=2;

(3)         q(x,y)
(4) {
(5)     y = a+x;
(6) }

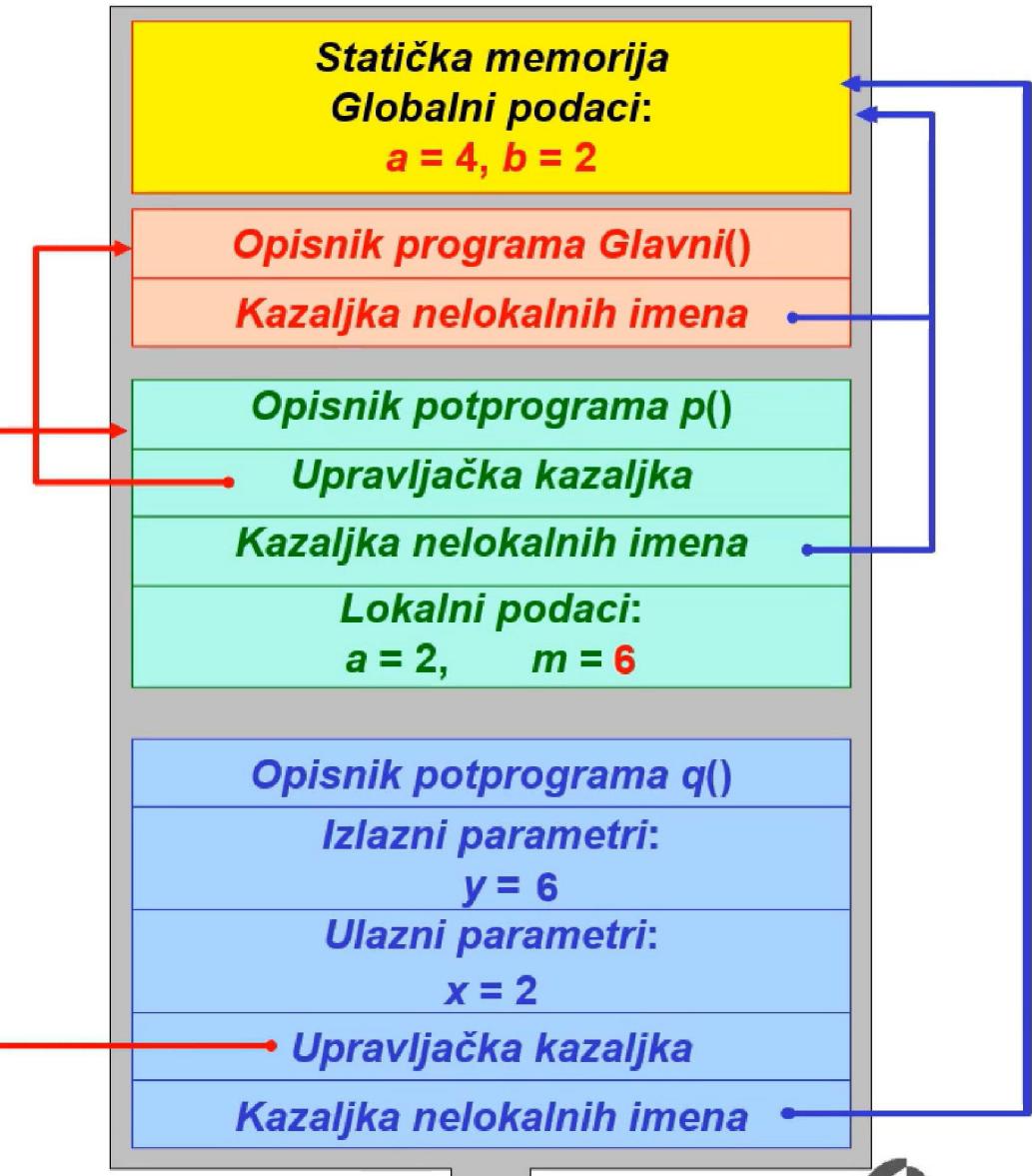
(7) p()
(8) int m=0, a=2;
(9) {
(10)    q(b,m);
(11) }

(12) {
(13)     p();
(14) }
```



Statičko pravilo djelokruga bez ugniježđenih procedura

```
(1) Glavni()
(2) int a=4, b=2;
     q()
     {
(3)         q(x,y)
(4)         {
(5)             y = a+x;
(6)         }
(7)         p()
(8)         int m=0, a=2;
(9)         {
(10)            q(b,m);
(11)        }
(12)    {
(13)        p();
(14)    }
```



Pristup nelokalnim imenima

- **Dinamičko pravilo djelokruga**
 - Temelji se na izvođenju programa:

Ako naredbe pozvane procedure ne deklariraju ime identifikatora,

onda su važeće deklaracije pozivajuće procedure

- Postupak određivanja važeće deklaracije
 - pretraživanje svih pozivajućih procedura
- Primjer
 - programski jezik Lisp

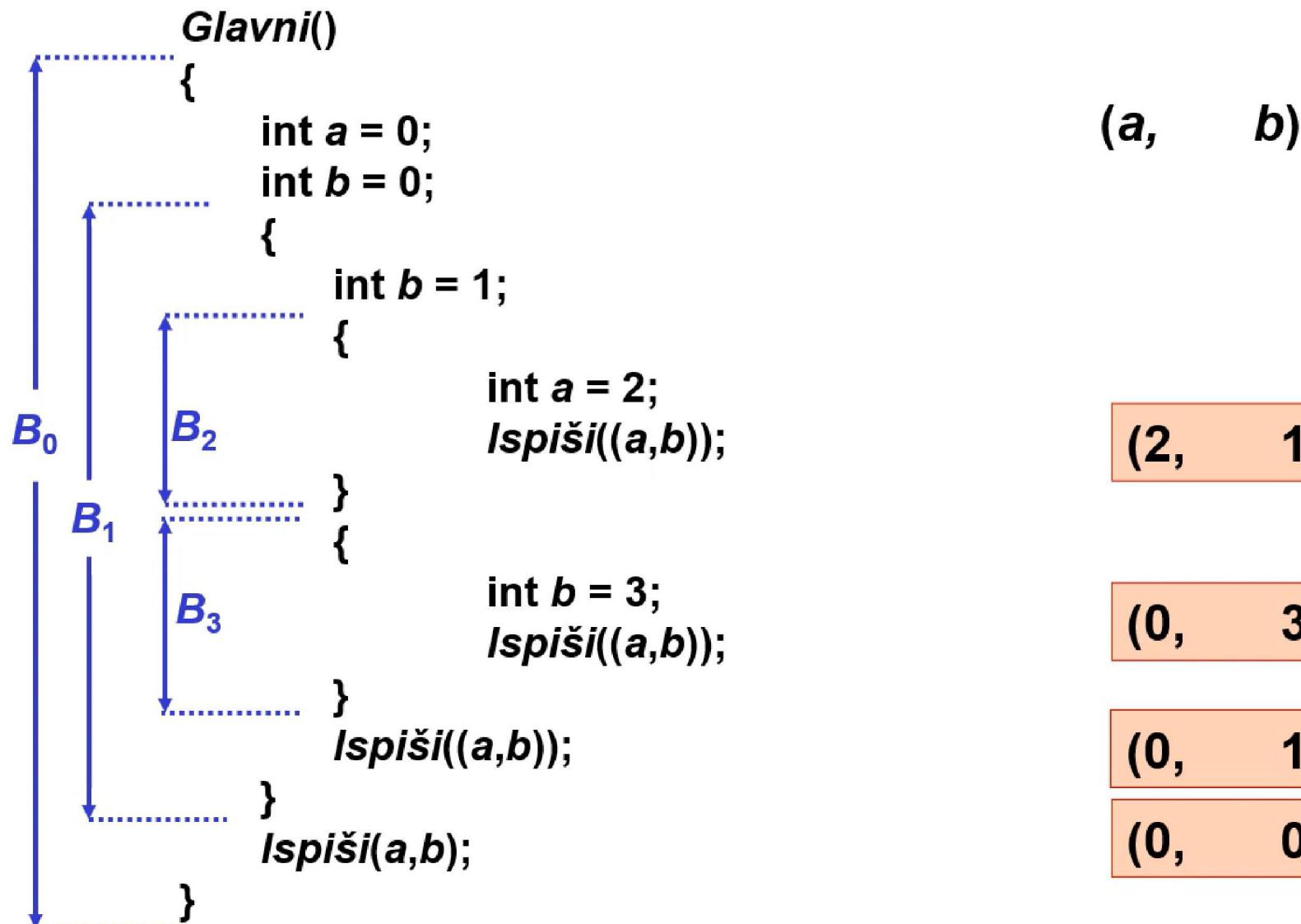


Pristup nelokalnim imenima

- **Statičko pravilo djelokruga ugniježđenih procedura**
 - **Primjer**
 - programski jezik Pascal
 - dozvoljava deklaraciju jedne procedure naredbama druge procedure
 - **Temelji se na pravilu najbliže ugnježđujuće procedure:**
 1. **Djelokrug deklaracije koja je zadana naredbom procedure uključuje sve naredbe te procedure.**
 2. **Ako ime x nije deklarirano naredbama procedure $q()$, onda je važeća deklaracija zadana naredbama ugnježđujuće procedure $p()$ za koju vrijedi (najbliže ugnježđujuća procedura):**
 - a) **Ime x deklarirano je naredbama procedure $p()$.**
 - b) **Ne postoji nijedna druga procedura $r()$ za koju vrijedi:**
 - **ime x deklarirano je naredbama procedura $r()$**
 - **procedura $p()$ ugnježđuje proceduru $r()$**
 - **procedura $r()$ ugnježđuje proceduru $q()$.**



Pristup nelokalnim imenima

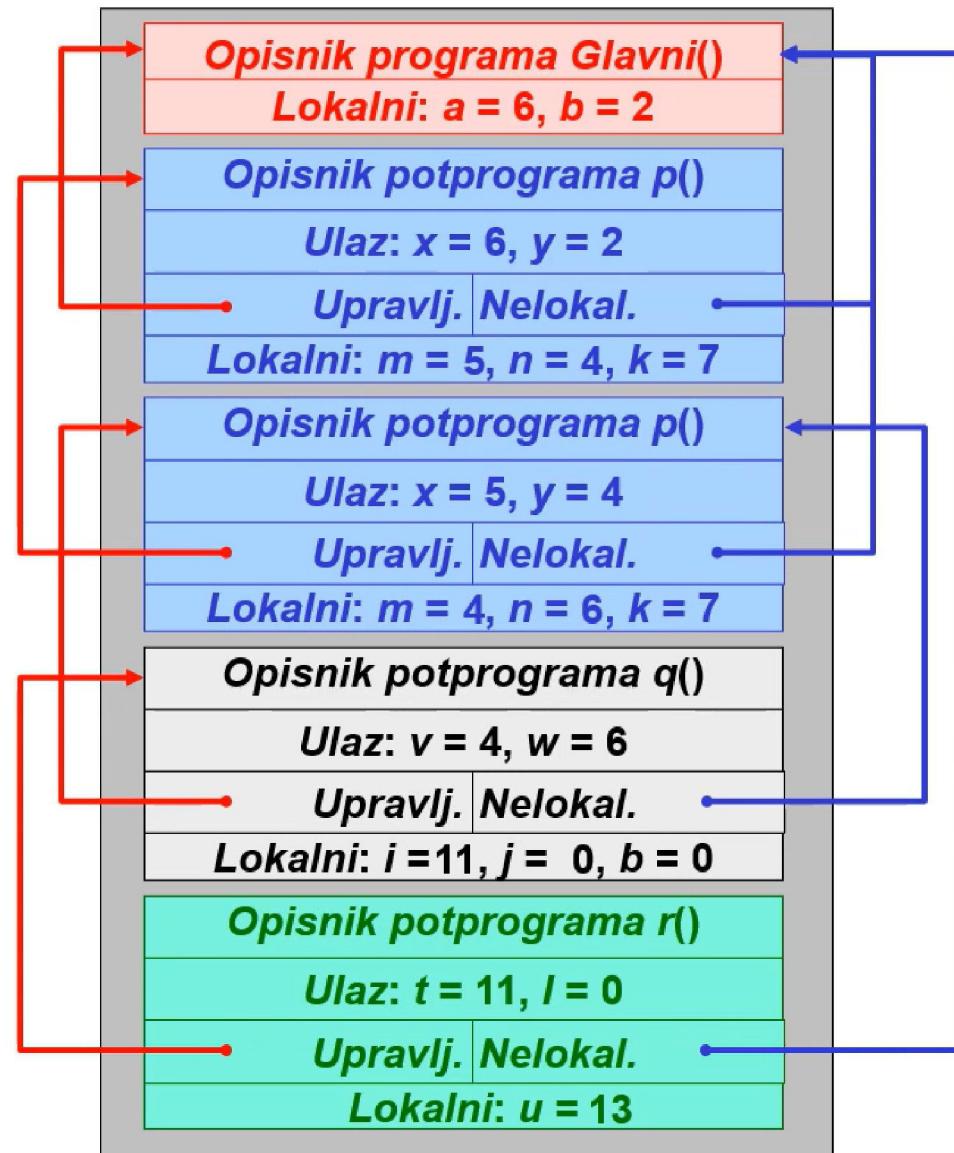


Statičko pravilo djelokruga ugniježđenih procedura

```

(1)  Glavni()
(2)  int a = 6, b = 2;
(3)    r(t,l);
(4)    int u;
(5)    {
(6)      u = t+b+l;
(7)    }
(8)    p(x,y)
(9)    int m, n, k = 7;
(10)   q(v,w)
(11)   int i,j,b = 0;
(12)   {
(13)     i = v+k+b;
(14)     j = w-a;
(15)     r(i,j);
(16)   }
(17)   {
(18)     m = x-1;
(19)     n = y+b;
(20)     ako ( m>=n )
(21)       p(m,n)
(22)     inače
(23)       q(m,n);
(24)   }
(25)   {
(26)     p(a,b);
(27)   }

```



Statičko pravilo djelokruga ugniježđenih procedura

- Neučinkovitost opisanog postupka traženja važeće deklaracije imena
 - pretraživanje opisnika ulančanih kazaljkama nelokalnih imena
- Dubina gniježđenja procedura
 - Dubina gniježđenja glavnog programa = 1
 - Ugniježđena procedura
 - dubina gniježđ. = dubina gniježđ. ugniježđujuće procedure + 1
 - Primjer
 - *Glavni()* = 1
 - potprogrami *r()* i *p()* = 2
 - potprogram *q()* = 3

Statičko pravilo djelokruga ugniježđenih procedura

- **Dubina gniježđenja identifikatora**
 - **Određuje se**
 - na temelju dubine gniježđenja procedure koja ga deklarira
 - **Za svaku uporabu identifikatora**
 - potrebno je zasebno odrediti dubinu gniježđenja
 - **Primjer**
 - dubina gniježđenja varijable *b* u retku (13) = 3
 - važeća deklaracija varijable *b* zadana je naredbama potprograma *q()* koji je dubine gniježđenja 3
 - dubina gniježđenja varijable *b* u retku (19) = 1
 - važeća deklaracija varijable *b* zadana je naredbama programa *Glavni()* koji je dubine gniježđenja 1
 - **Dubina gniježđenja**
 - izračuna se tijekom prevodenja izvornog programa u ciljni program
 - koristi se tijekom izvođenja ciljnog programa



Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju

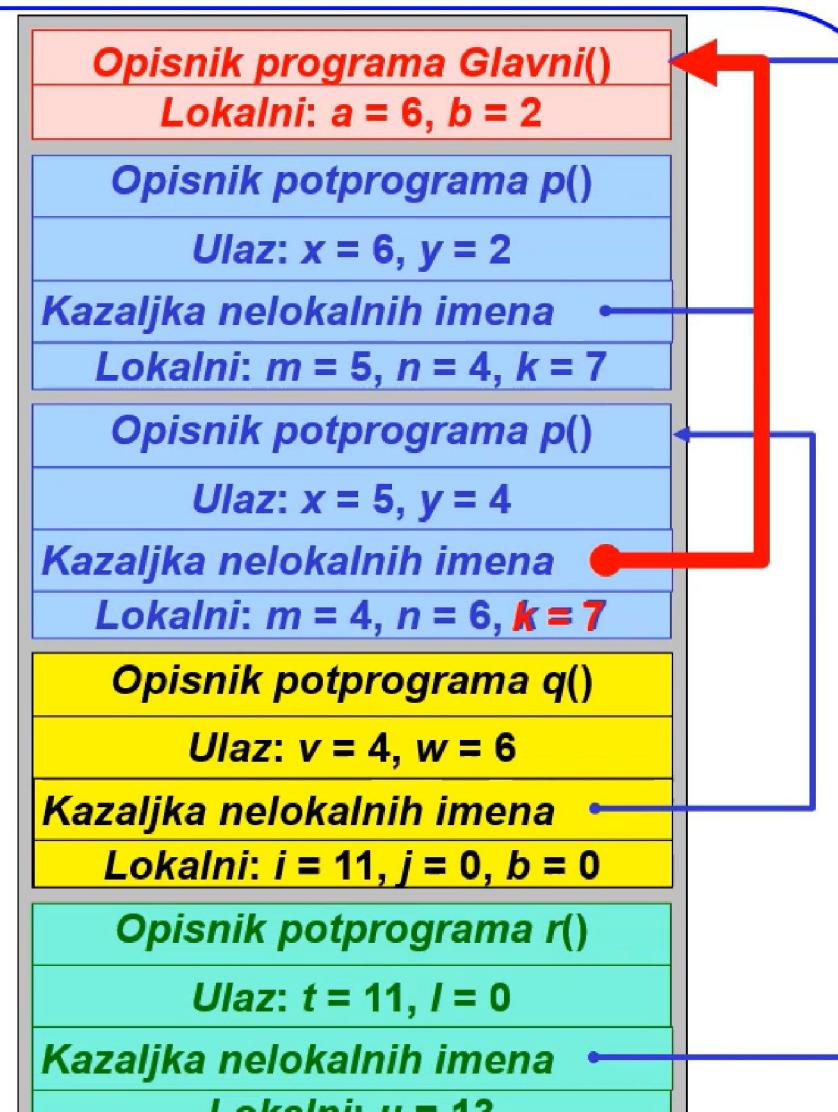
- **Nelokalno ime a**
 - dubine gniježđenja n_a
 - koristi se u proceduri $p()$
 - dubine gniježđenja n_p
 - $n_a < n_p$
- **Dohvat opisnika procedure koja deklarira ime a**
 - Slijedi se $n_p - n_a$ kazaljki nelokalnih imena počevši od kazaljke zapisane u opisniku procedure $p()$
- **Nelokalno ime u opisniku procedure $p()$ određeno je sljedećim vrijednostima:**
 - $(n_p - n_a, \text{PomakOdPočetneAdreseOpisnika})$
 - **Obje vrijednosti**
 - moguće je odrediti tijekom prevodenja izvornog u ciljni program
 - **Nije potrebno pretraživati sadržaje opisnika koji su ulančani kazaljkama nelokalnih imena**



```

(1)  Glavni()
(2)    int a = 6, b = 2;
(3)    r(t,l);
(4)    int u;
(5)    {
(6)      u = t+b+l;
(7)    }
(8)    p(x,y)
(9)    int m, n, k = 7;
(10)   q(v,w)
(11)   int i,j,b = 0;
(12)   {
(13)     i = v + k + b;
(14)     j = w - a;
(15)     r(i,j);
(16)   }
(17)   {
(18)     m = x-1;
(19)     n = y+b;
(20)     ako ( m>=n )
(21)       p(m,n)
(22)     inače
(23)       q(m,n);
(24)   }
(25)   {
(26)     p(a,b);
(27)   }

```



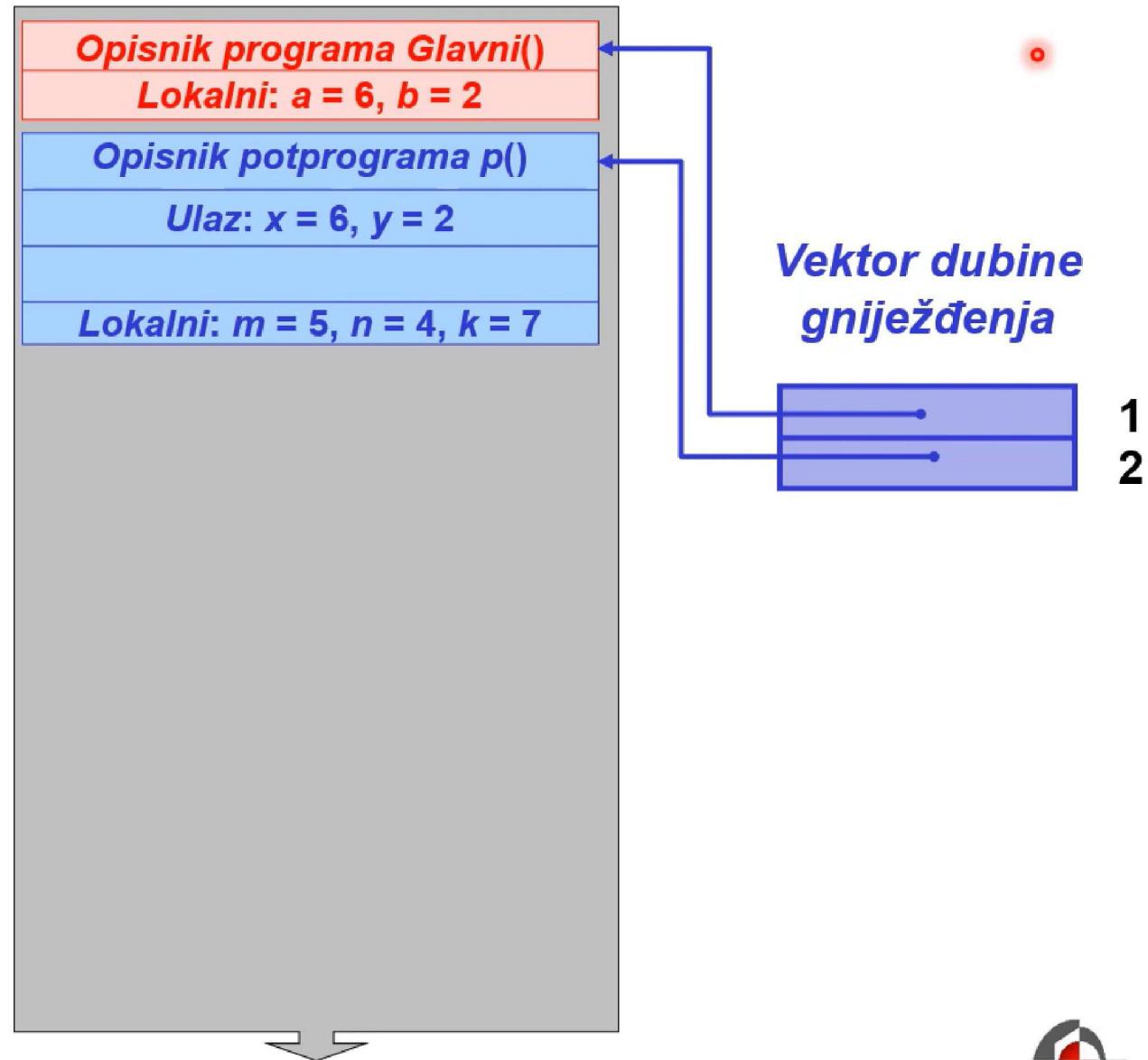
$$n_q = 3 \quad n_k = 2 \quad n_a = 1$$

$$n_q - n_k = 3 - 2 = 1 \quad n_q - n_a = 3 - 1 = 2$$

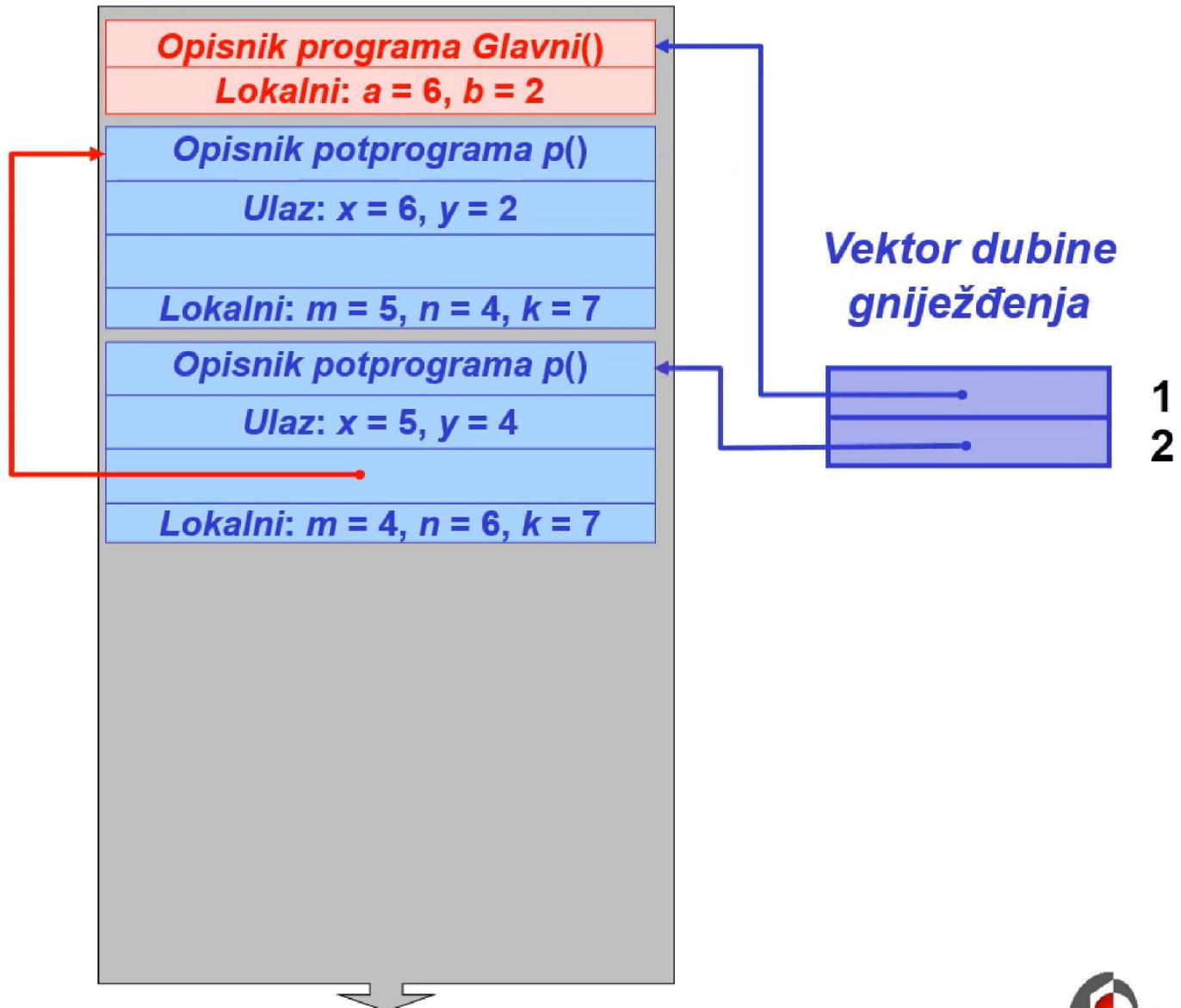
Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju

- **Neučinkovitost**
 - čitanje vrijednosti ulančanih kazaljki nelokalnih imena
- **Vektor dubine gniježđenja**
 - **Dubina gniježđenja procedure koja se izvodi = n**
 - vektor ima n elemenata
 - **Dubina gniježđenja i , $1 \leq i \leq n$**
 - dodjeljuje se element vektora i
 - element vektora i
 - kazaljka koja pokazuje na opisnik posljednje aktivirane procedure dubine gniježđenja i
 - izravno dohvaćanje opisnika koji sadrži deklaraciju zadano imena a

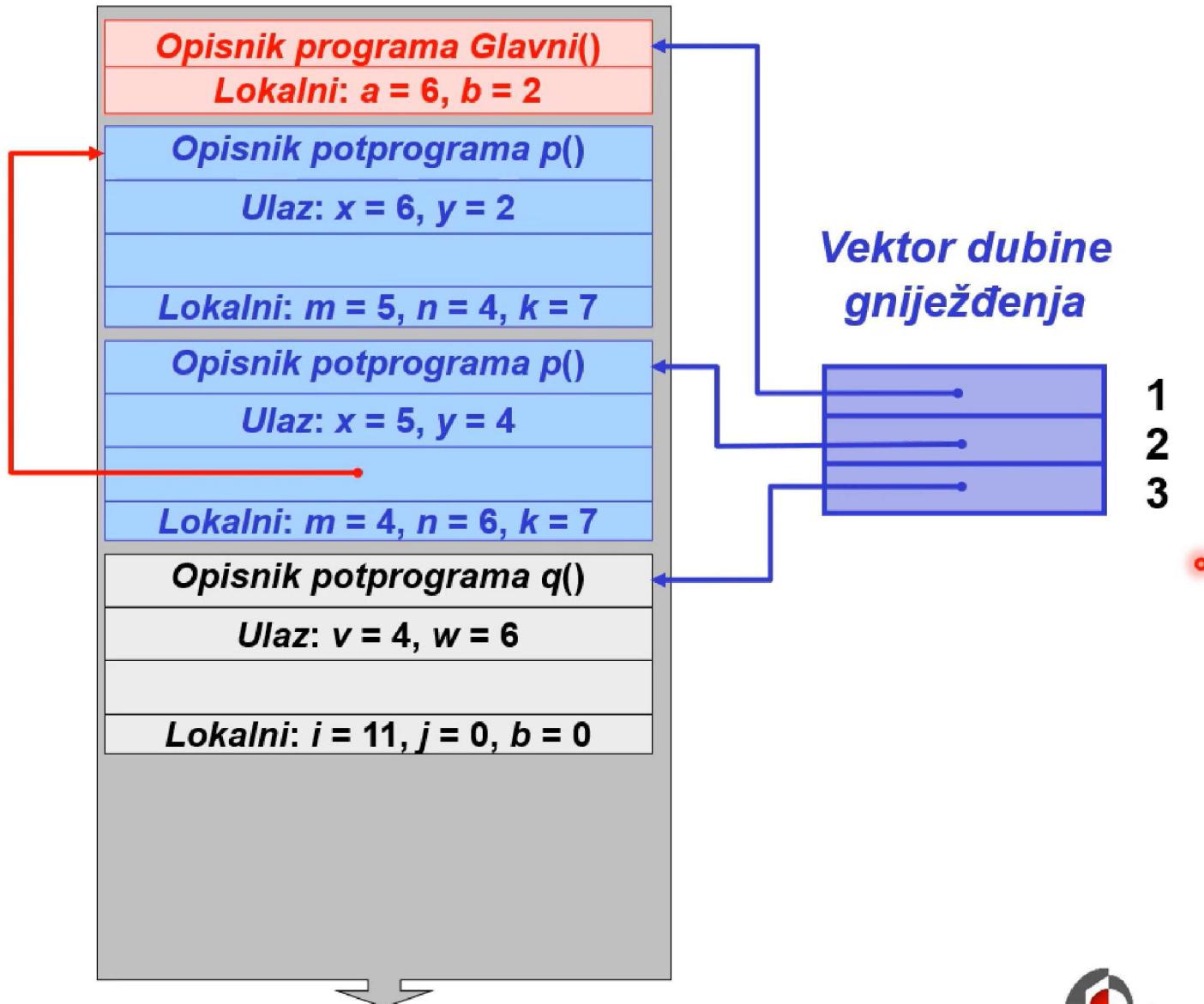
Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



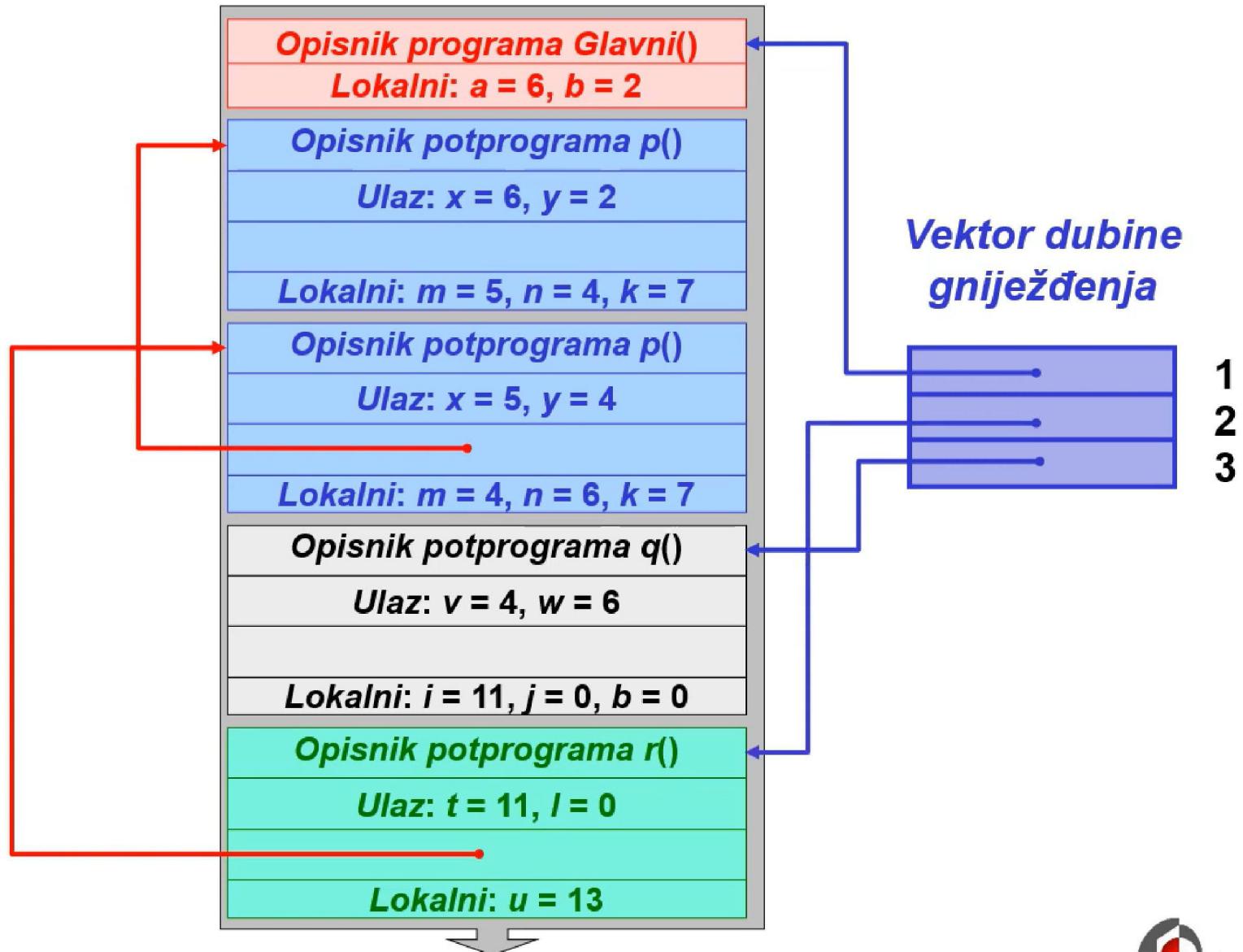
Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



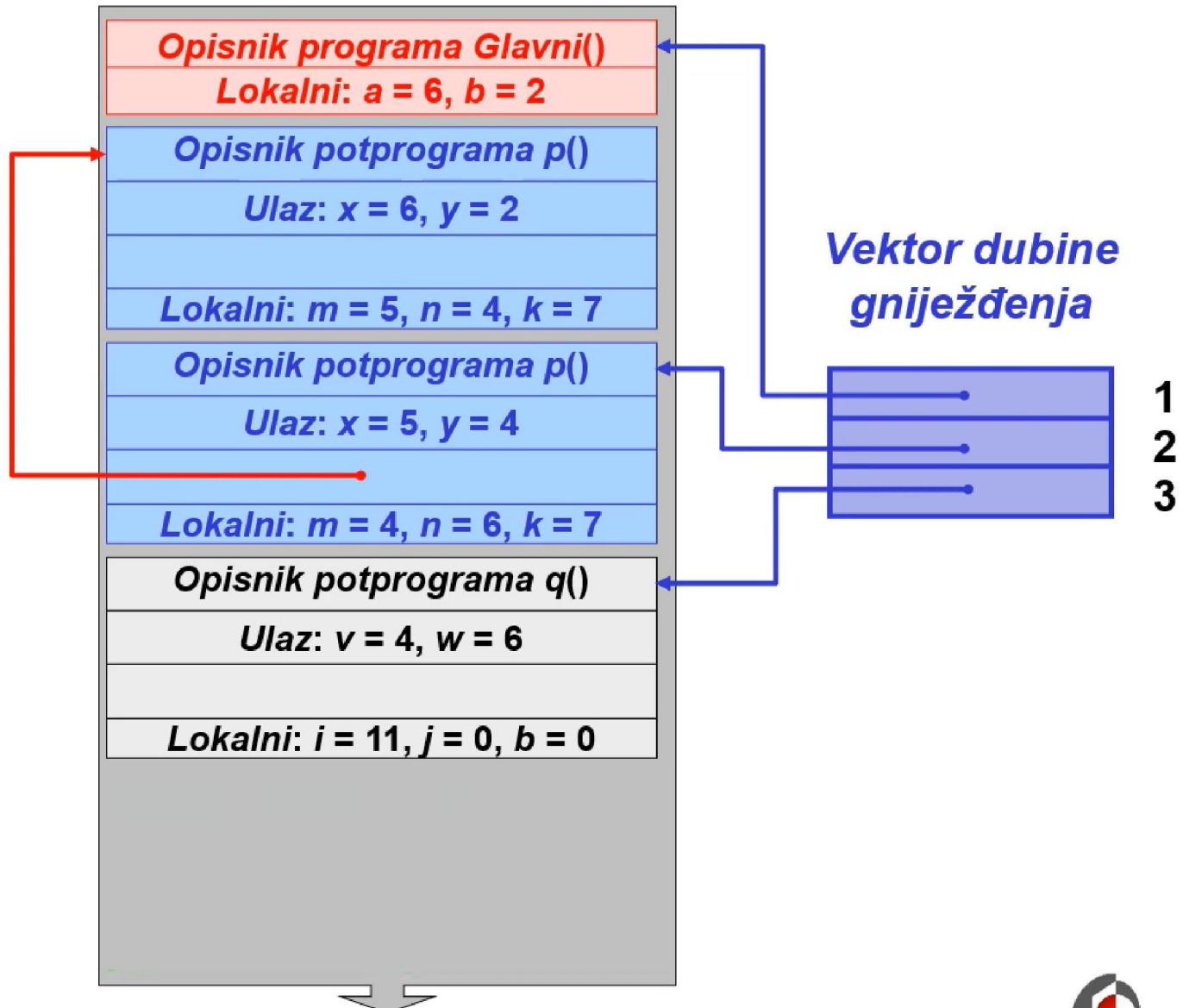
Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



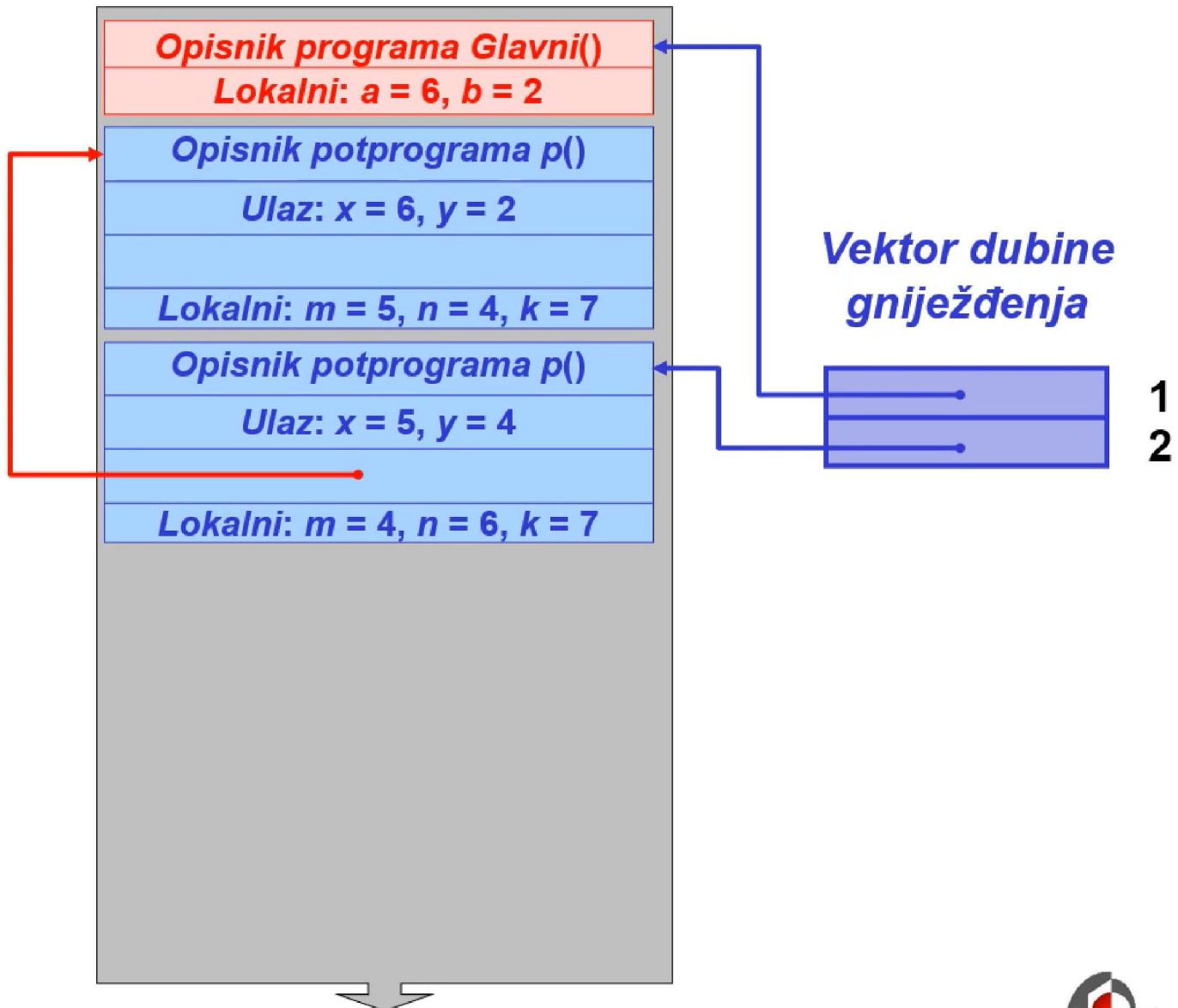
Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



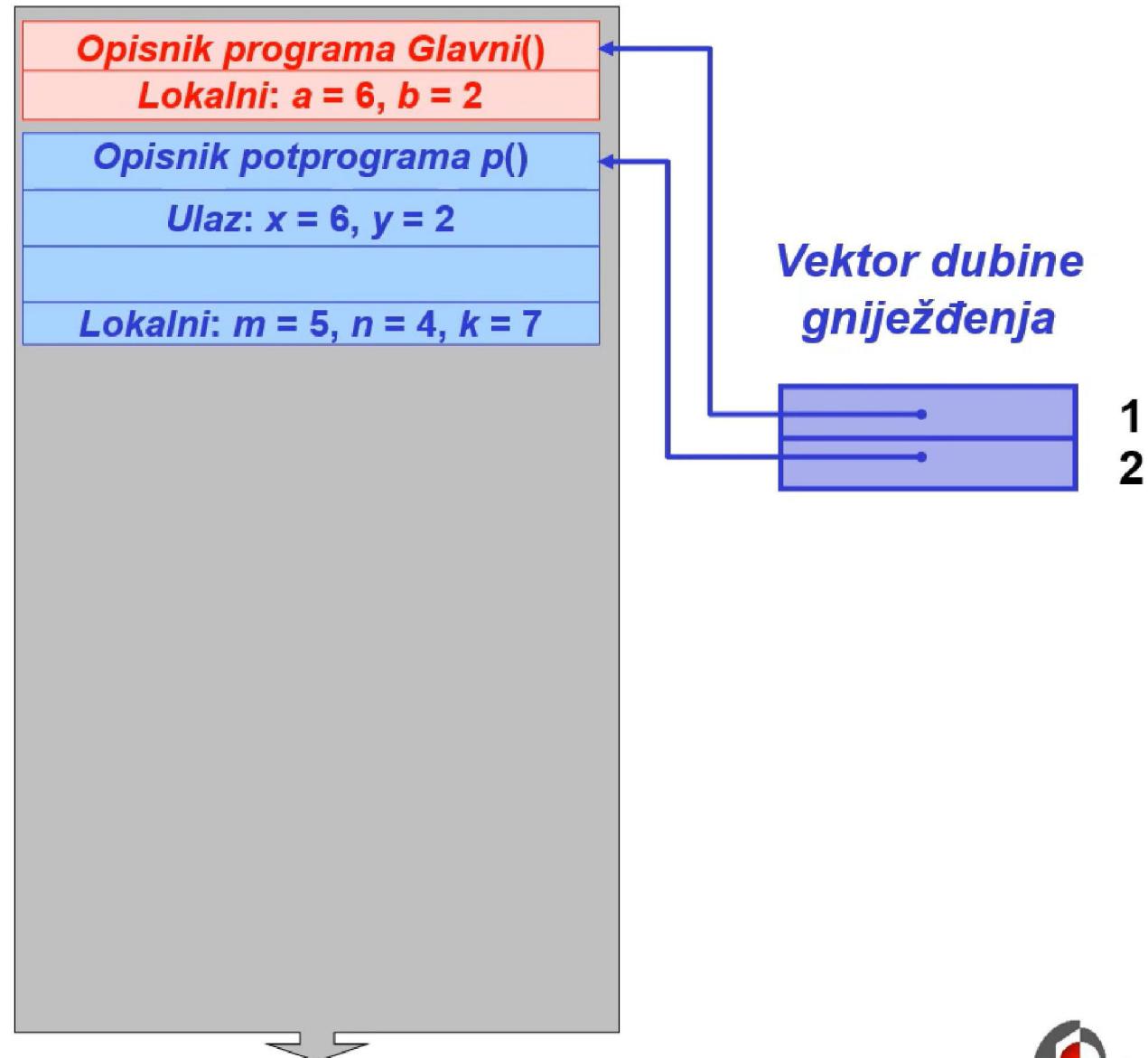
Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



Algoritam traženja opisnika procedure koji sadrži važeću deklaraciju



Algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja

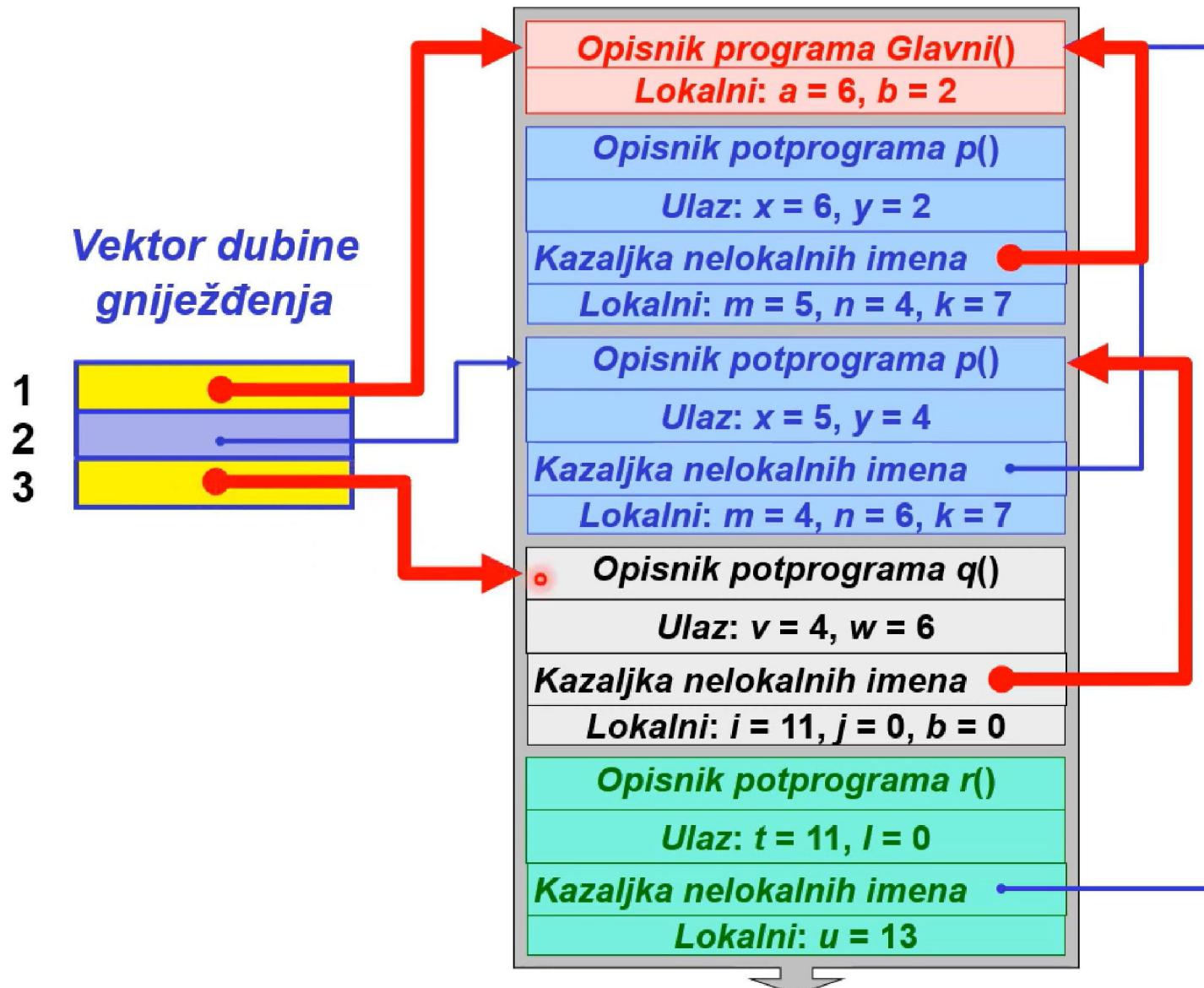
- **Dva različita slučaja**
 - **Pozvana procedura deklarirana je naredbama pozivajuće procedure**
 - Dubina gniježđenja pozvane procedure i za jedan je veća od dubine gniježđenja pozivajuće procedure j ($i = j + 1$)
 - **Pozvana procedura deklarirana je naredbama procedure koja ugnježđuje pozvanu i pozivajuću proceduru**
 - Dubina gniježđenja pozvane procedure i jednaka je ili manja od dubine gniježđenja pozivajuće procedure j ($i \leq j$)



Algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja

- Pozvana procedura deklarirana je naredbama pozivajuće procedure
 - Dubina gniježđenja pozvane procedure i za jedan je veća od dubine gniježđenja pozivajuće procedure j ($i = j + 1$)
 - Kazaljka nelokalnih imena pozvane procedure
 - pokazuje na opisnik pozivajuće procedure
 - Vektor dubine gniježđenja
 - doda se novi element $i = j + 1$ koji pokazuje na opisnik pozvane procedure

Algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja



Algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja

- Pozvana procedura deklarirana je naredbama procedure koja ugnježđuje pozvanu i pozivajuću proceduru
 - Dubina gniježđenja pozvane procedure i jednaka je ili manja od dubine gniježđenja pozivajuće procedure j ($i \leq j$)
 - Kazaljka nelokalnih imena pozvane procedure
 - Pokazuje na opisnik procedure koji se dohvati slijedenjem $j-(i-1)$ kazaljki nelokalnih imena počevši od kazaljke pozivajuće procedure
 - Dubina gniježđenja pozvane procedure i
 - dubina njezine najbliže ugnježđujuće procedure jest $i-1$
 - Vektor dubine gniježđenja
 - Vrijednost elementa vektora na mjestu i
 - spremi se u opisnik pozvane procedure
 - Element vektora na mjestu i
 - kazaljka na opisnik pozvane procedure

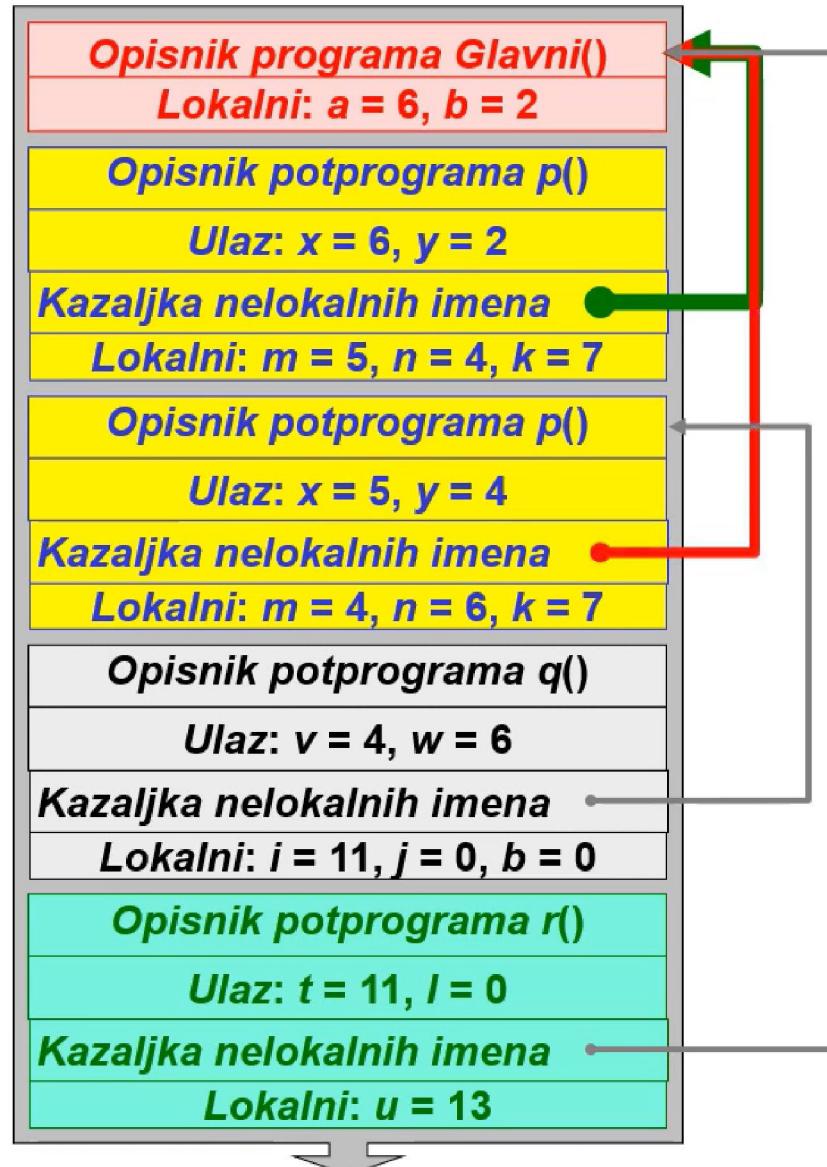


Algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja

Dubina gniježđenja
pozivajuće procedure $j = 2$

Dubina gniježđenja
pozvane procedure $i = 2$

$$j-(i-1) = 2-(2-1) = 1$$

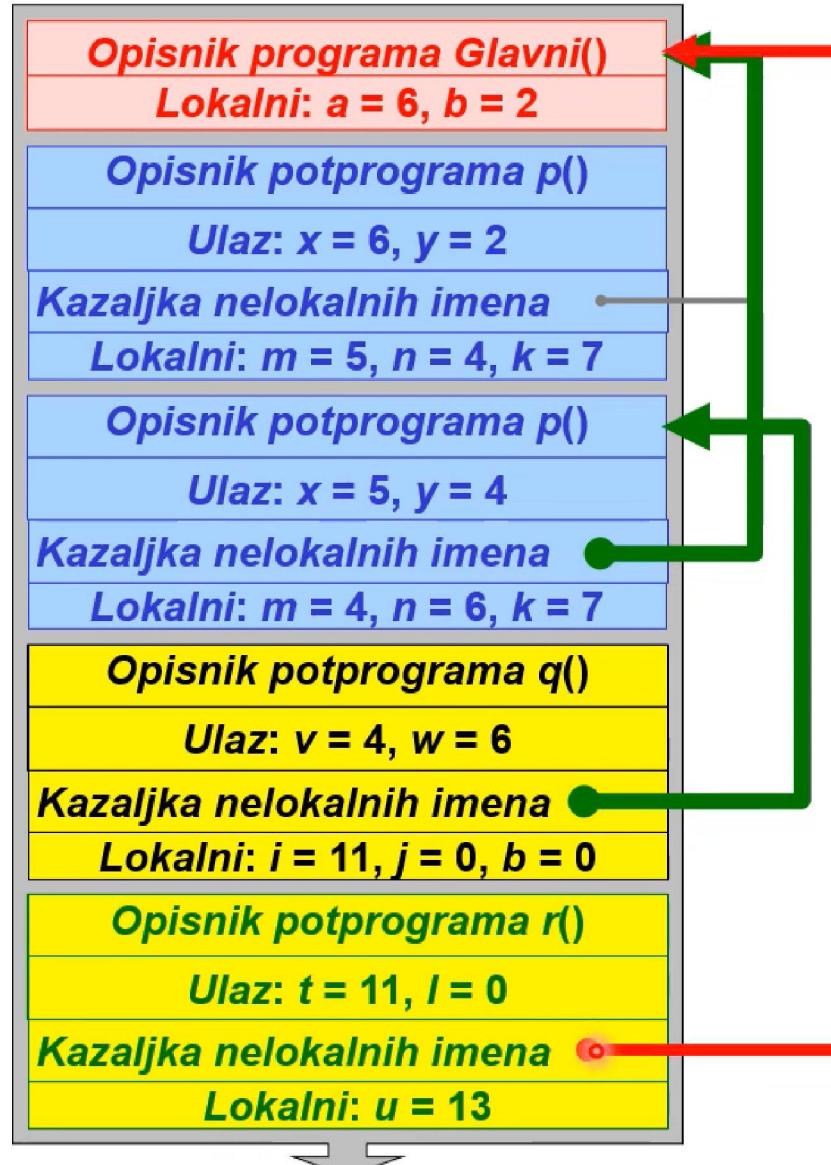


Algoritam gradnje lanca kazaljki nelokalnih imena i vektora dubine gniježđenja

Dubina gniježđenja
pozivajuće procedure $j = 3$

Dubina gniježđenja
pozvane procedure $i = 2$

$$j - (i-1) = 3 - (2-1) = 2$$

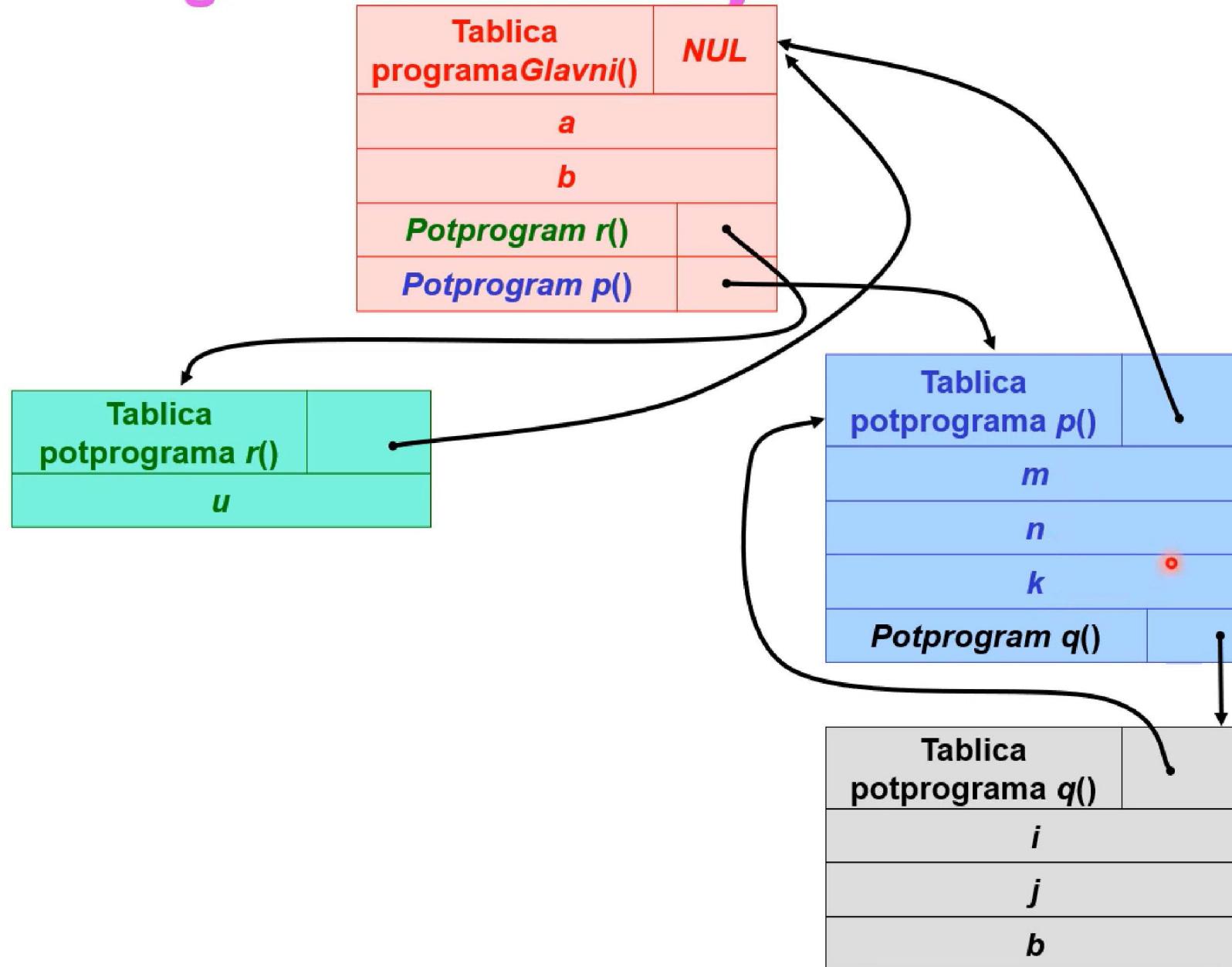


Programsko ostvarenje tablice znakova

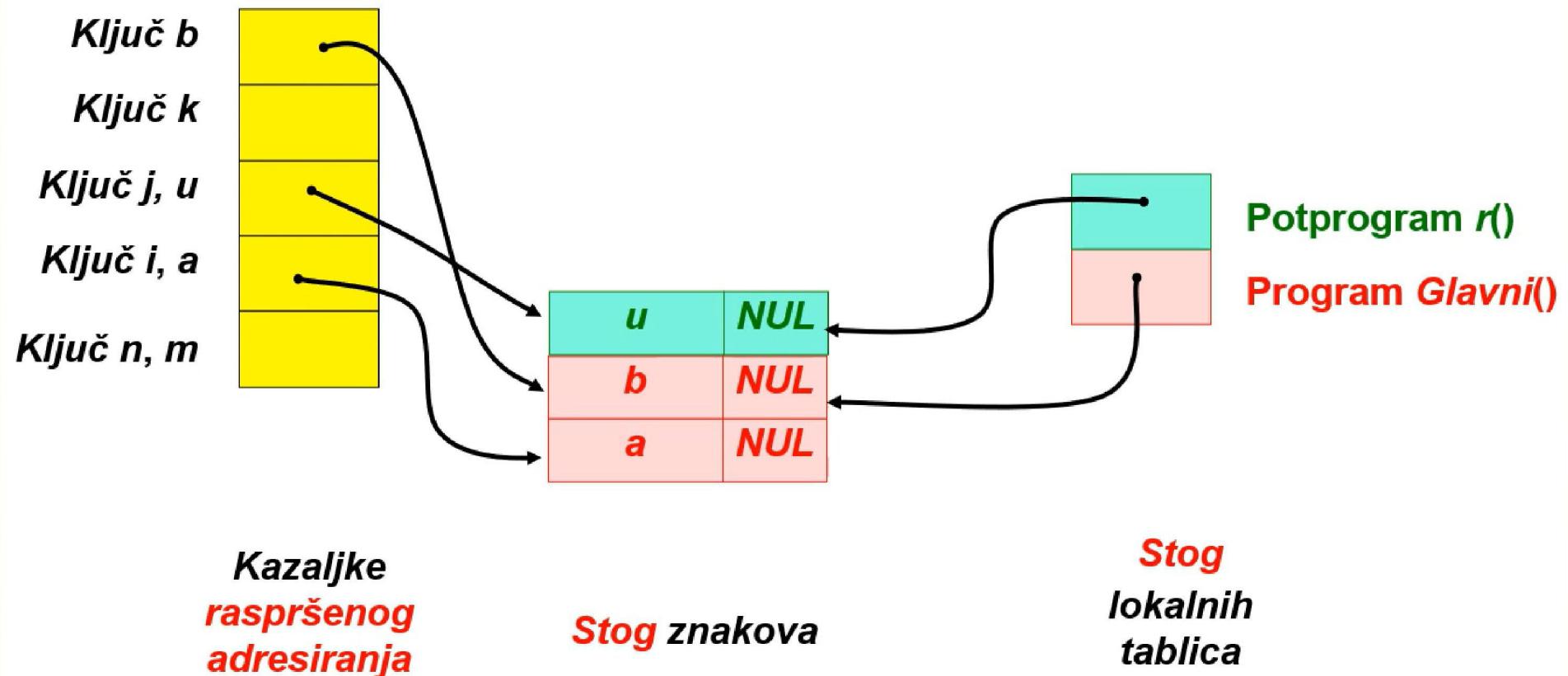
- **Statičko pravilo ugniježđenih procedura**
 - **Tablica znakova – stablo**
 - Čvorovi stabla
 - lokalne tablice znakova pojedinih procedura
 - Korijen stabla
 - lokalna tablica znakova glavnog programa
 - Procedura *A()* ugnježđuje proceduru *B()*
 - čvor *A()* roditelj čvora *B()*
 - Procedura *A()* deklarirana ispred procedure *B()*
 - čvor *A()* lijevo od čvora *B()*



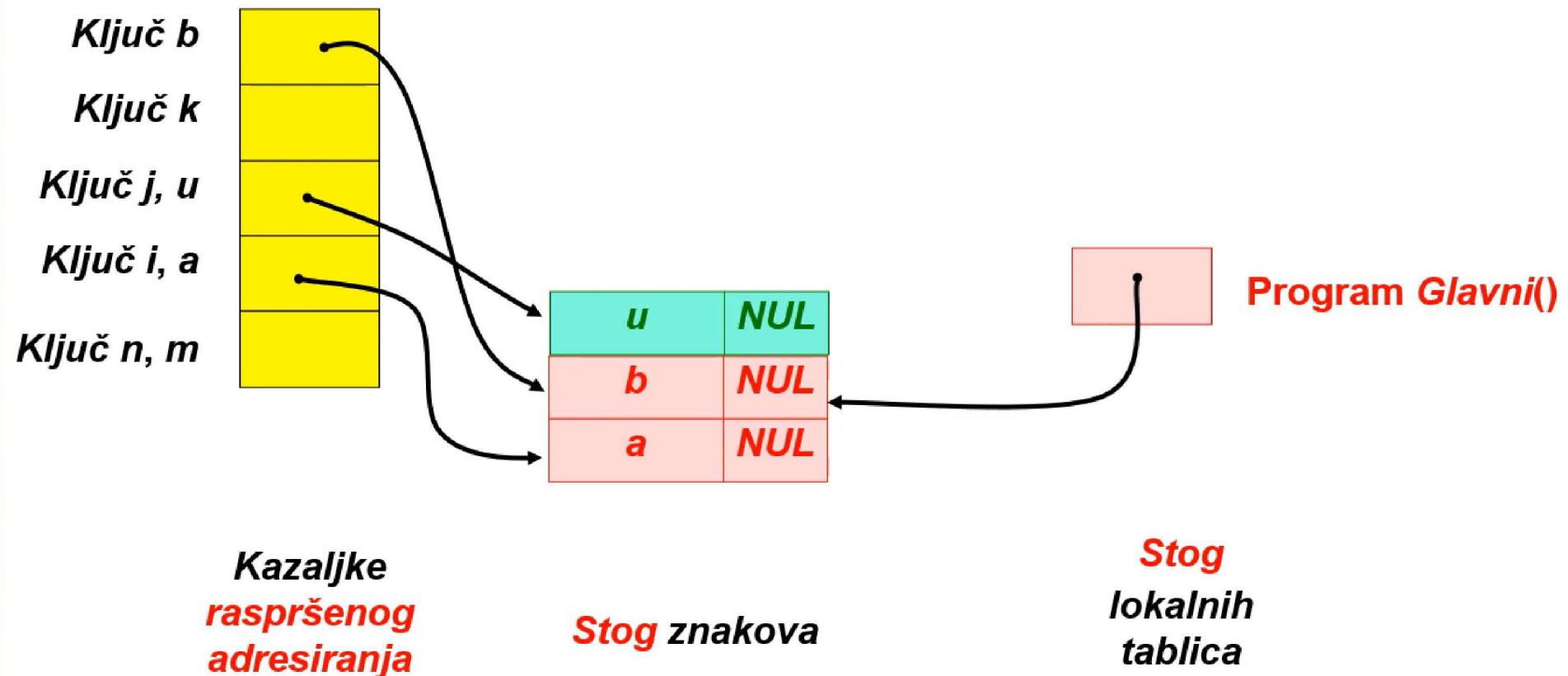
Programsko ostvarenje tablice znakova



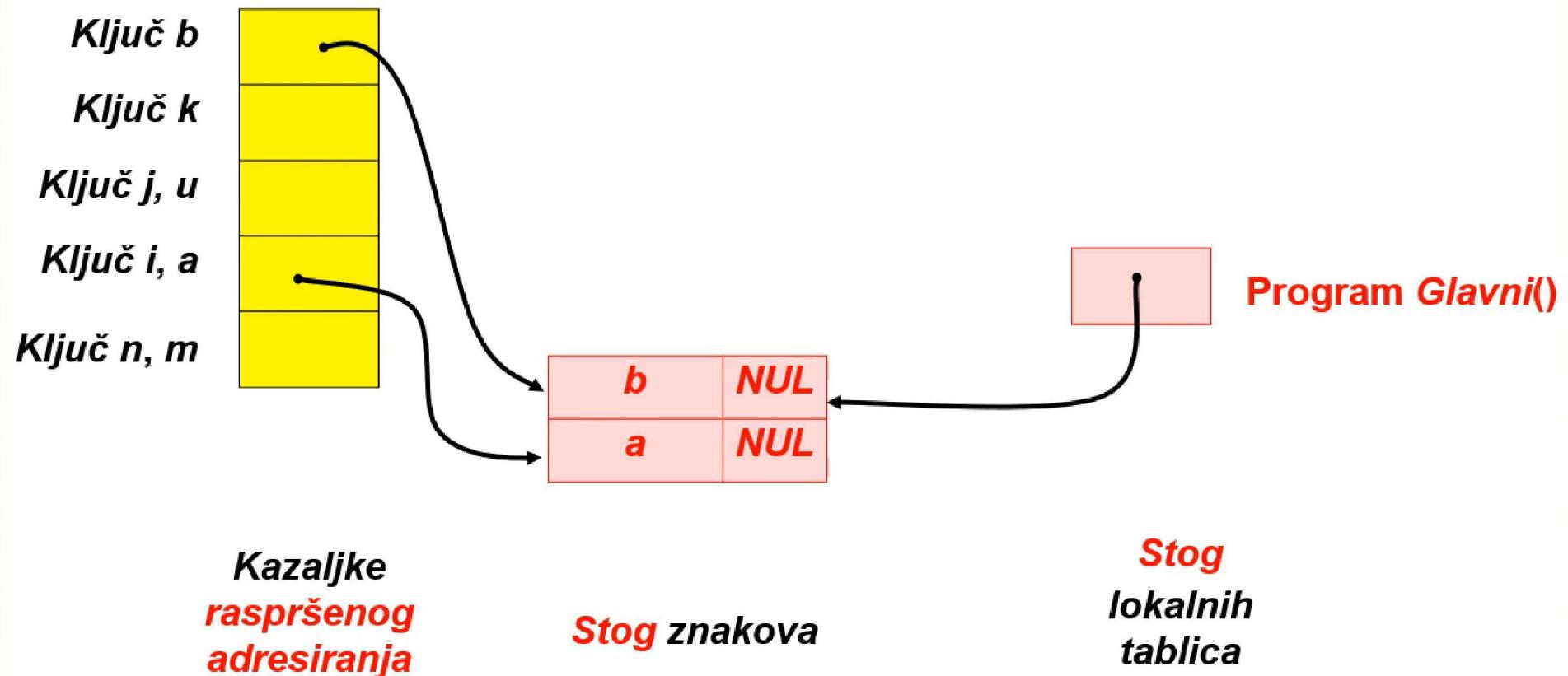
Programsko ostvarenje tablice znakova



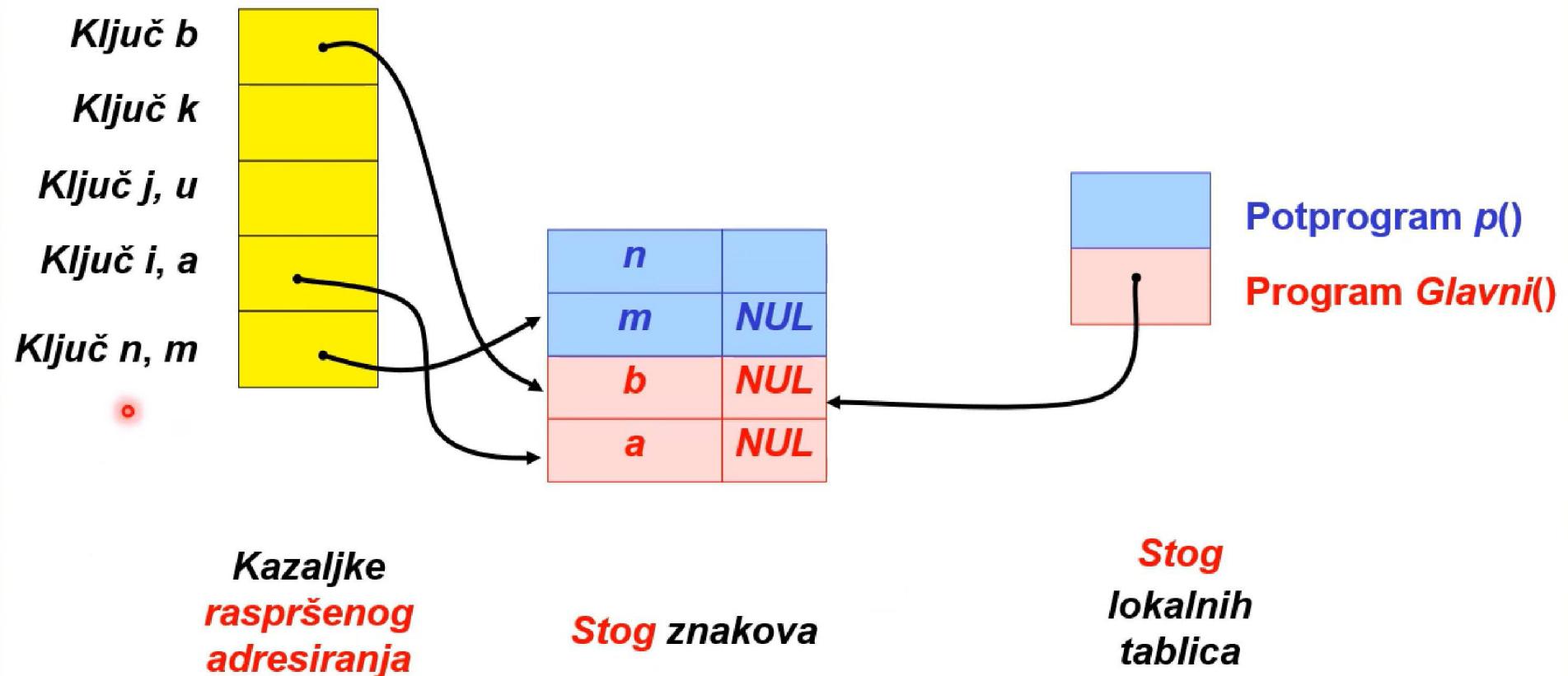
Programsko ostvarenje tablice znakova



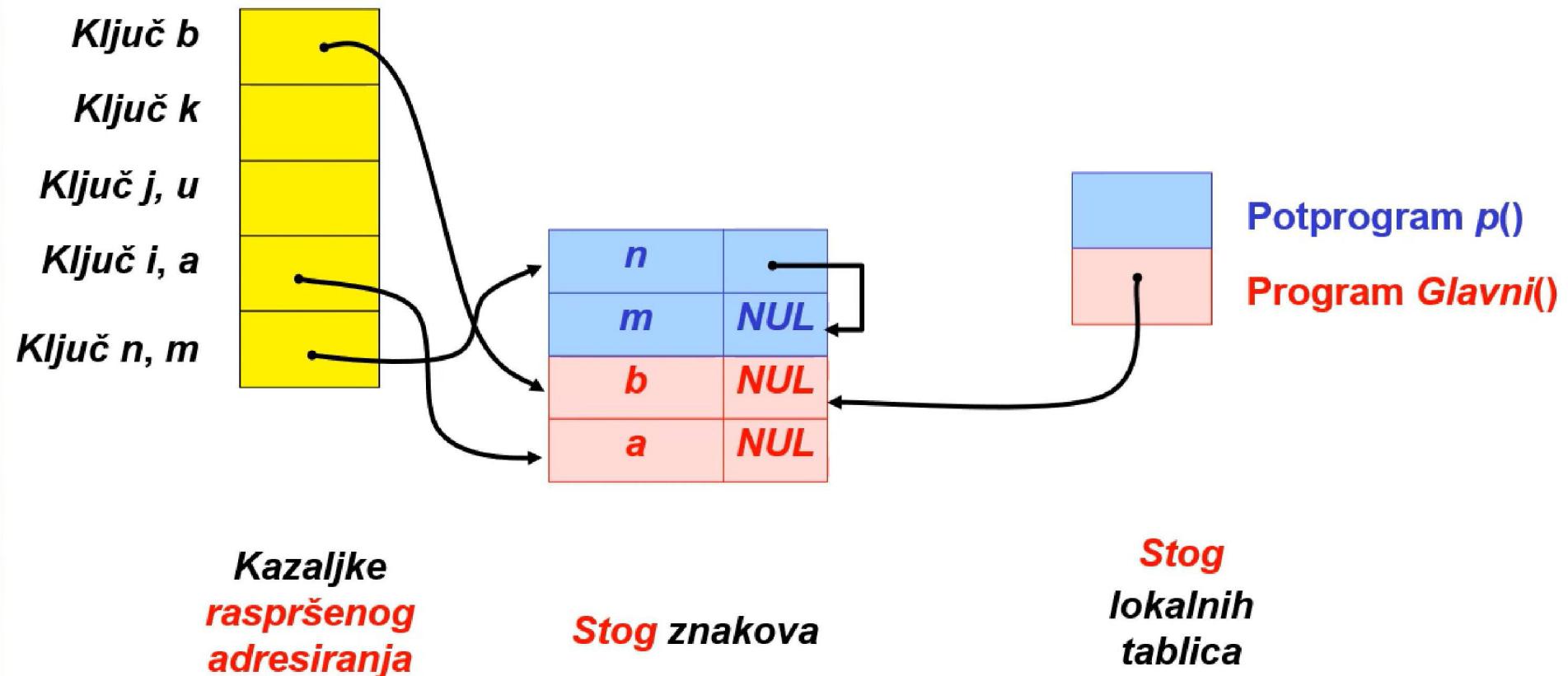
Programsko ostvarenje tablice znakova



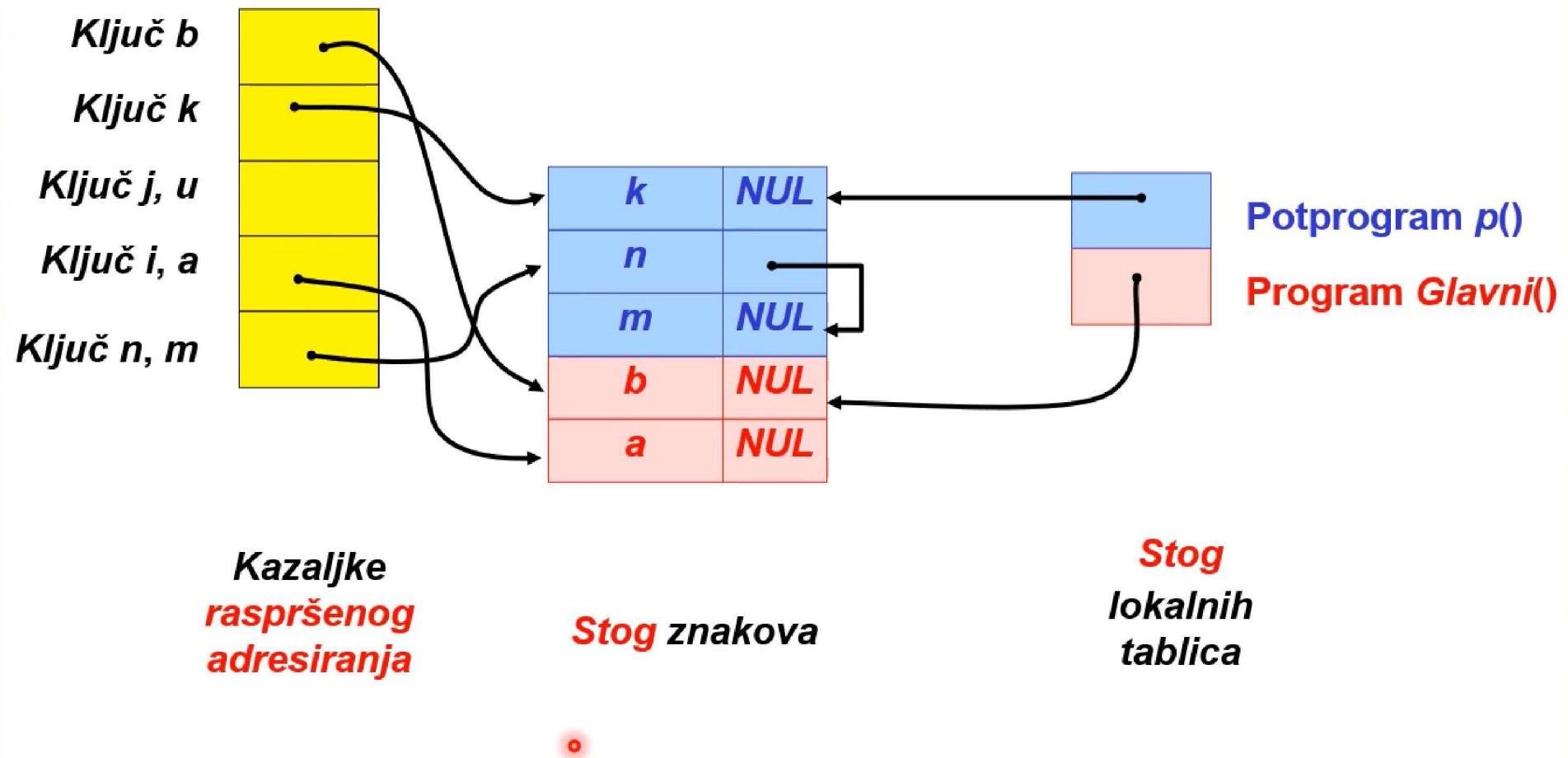
Programsko ostvarenje tablice znakova



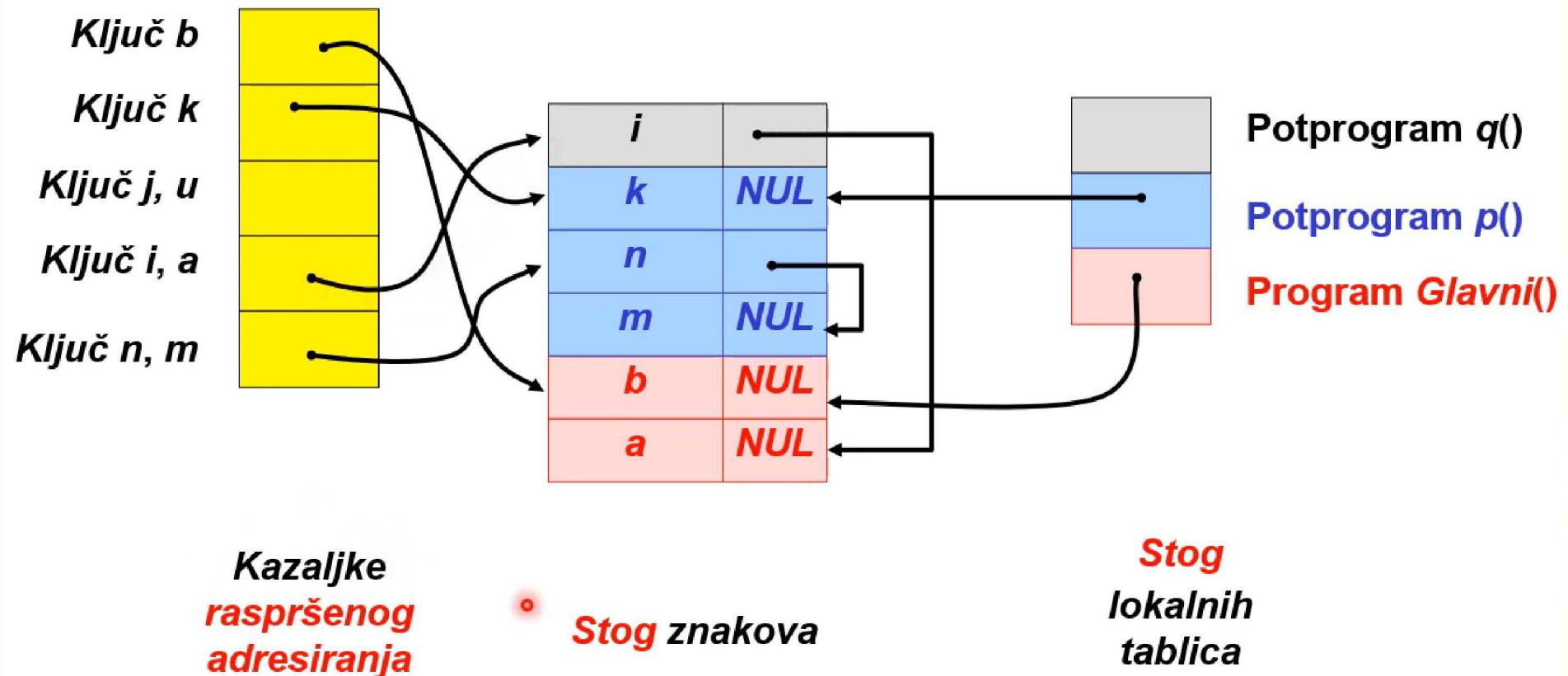
Programsko ostvarenje tablice znakova



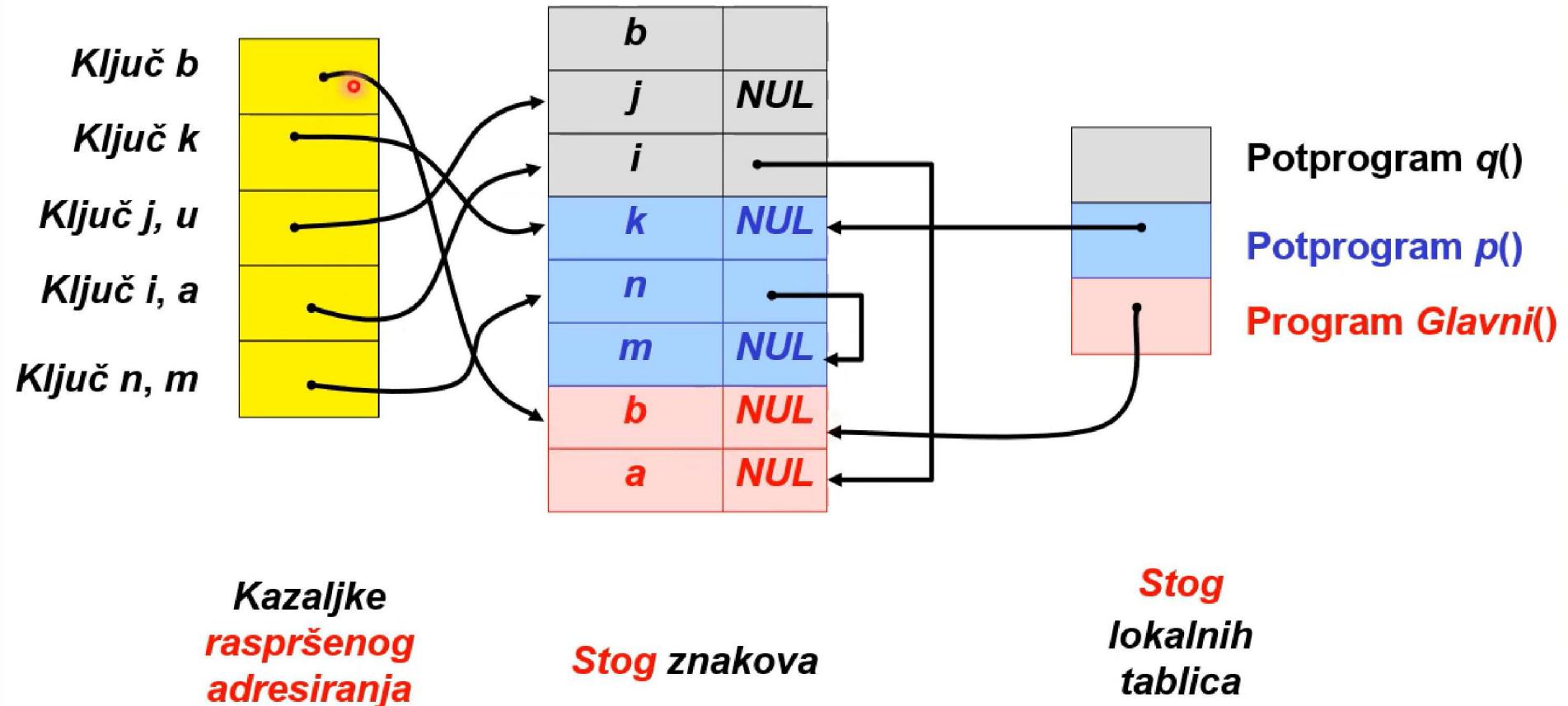
Programsko ostvarenje tablice znakova



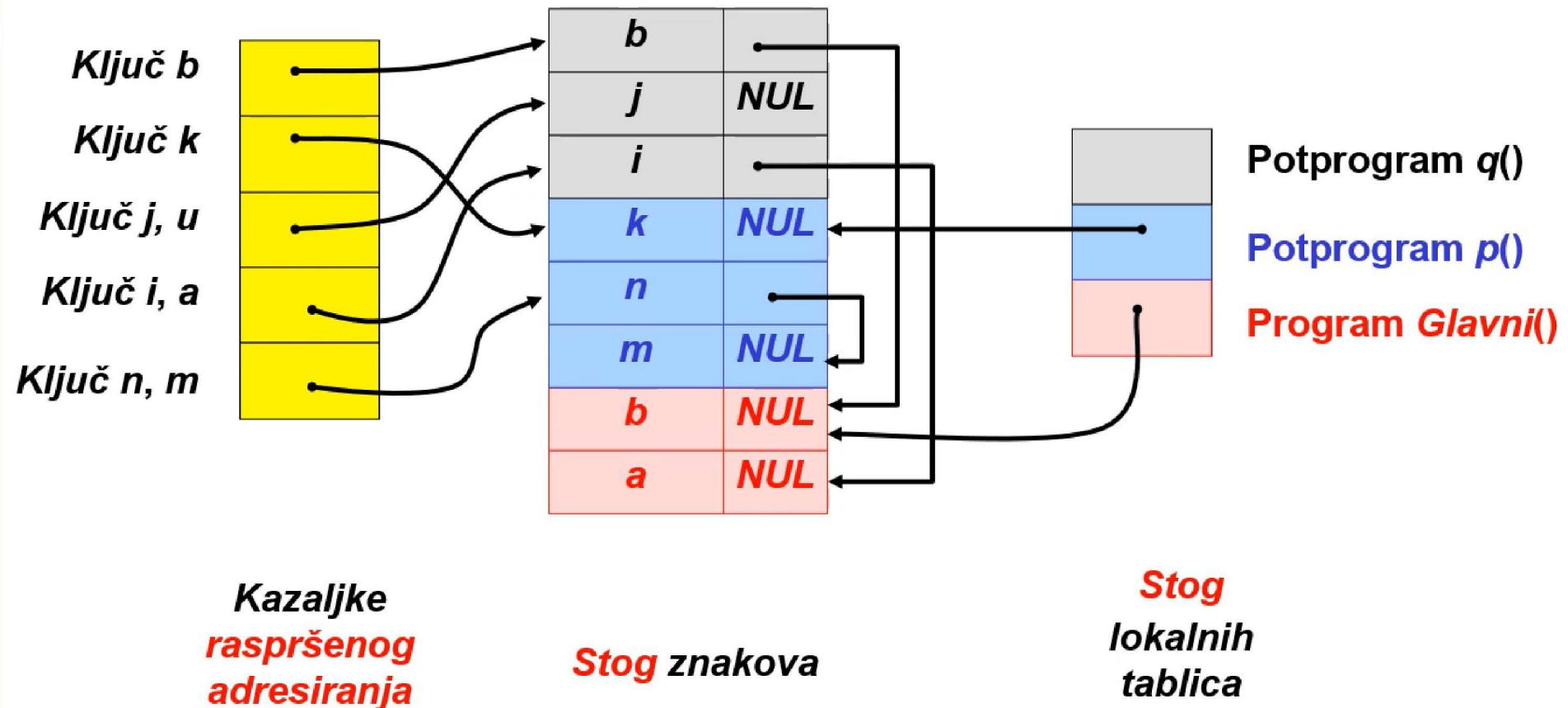
Programsko ostvarenje tablice znakova



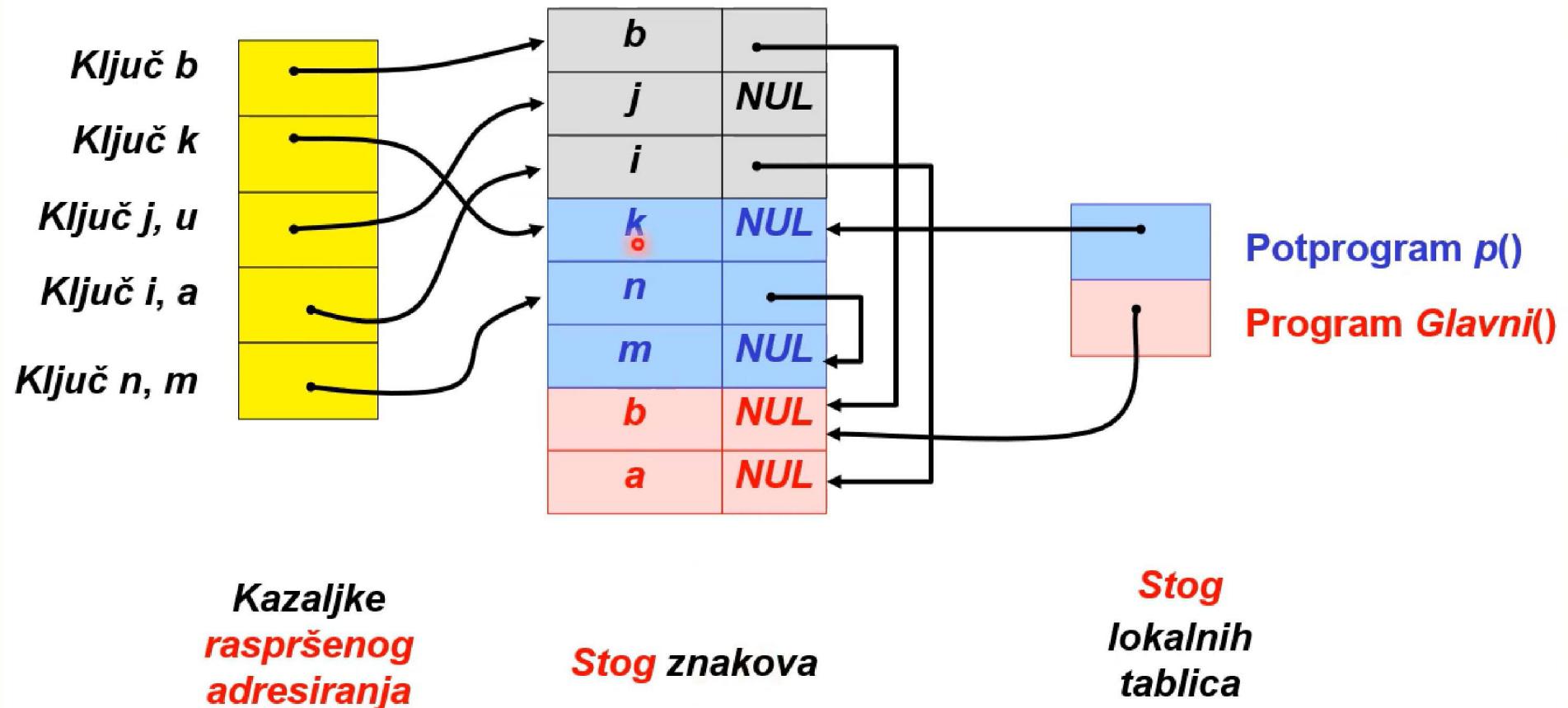
Programsko ostvarenje tablice znakova



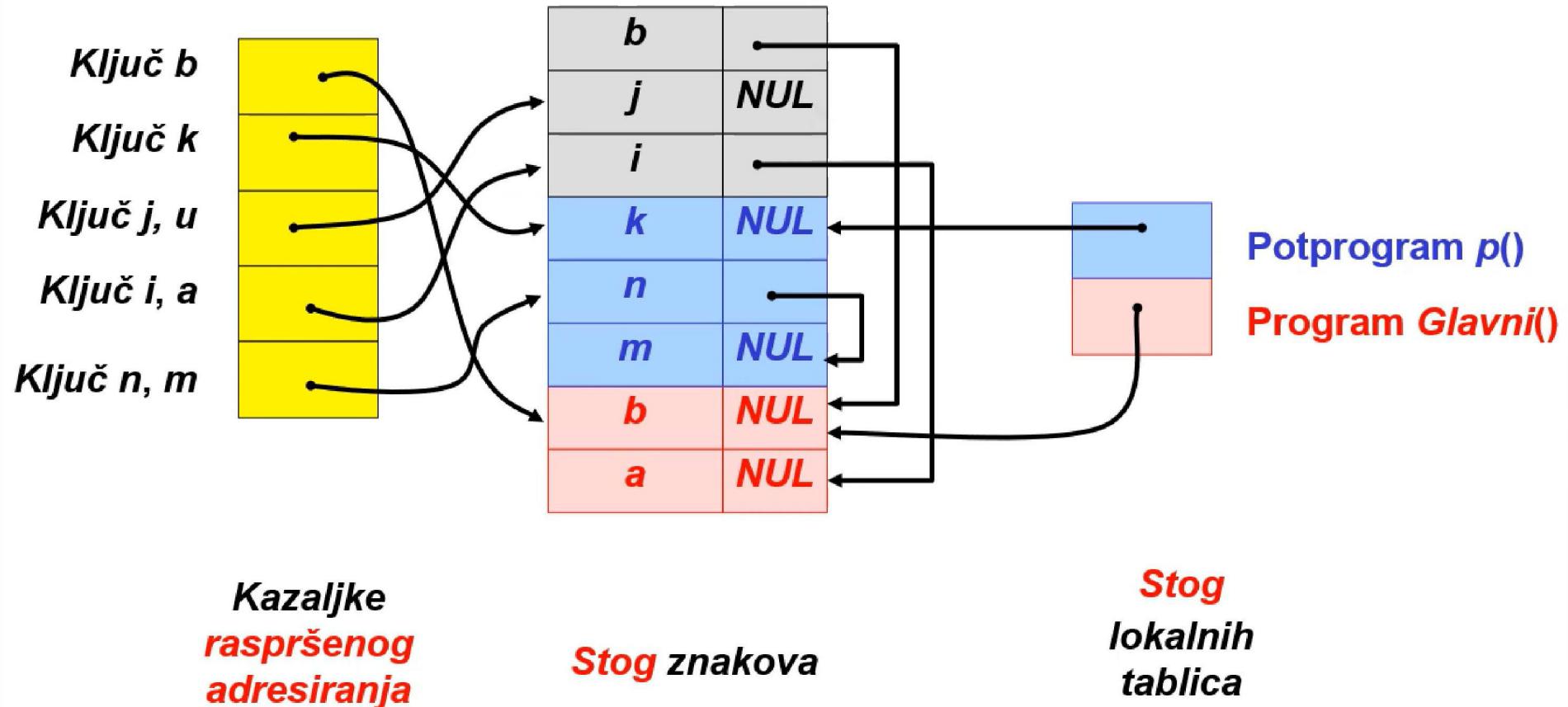
Programsko ostvarenje tablice znakova



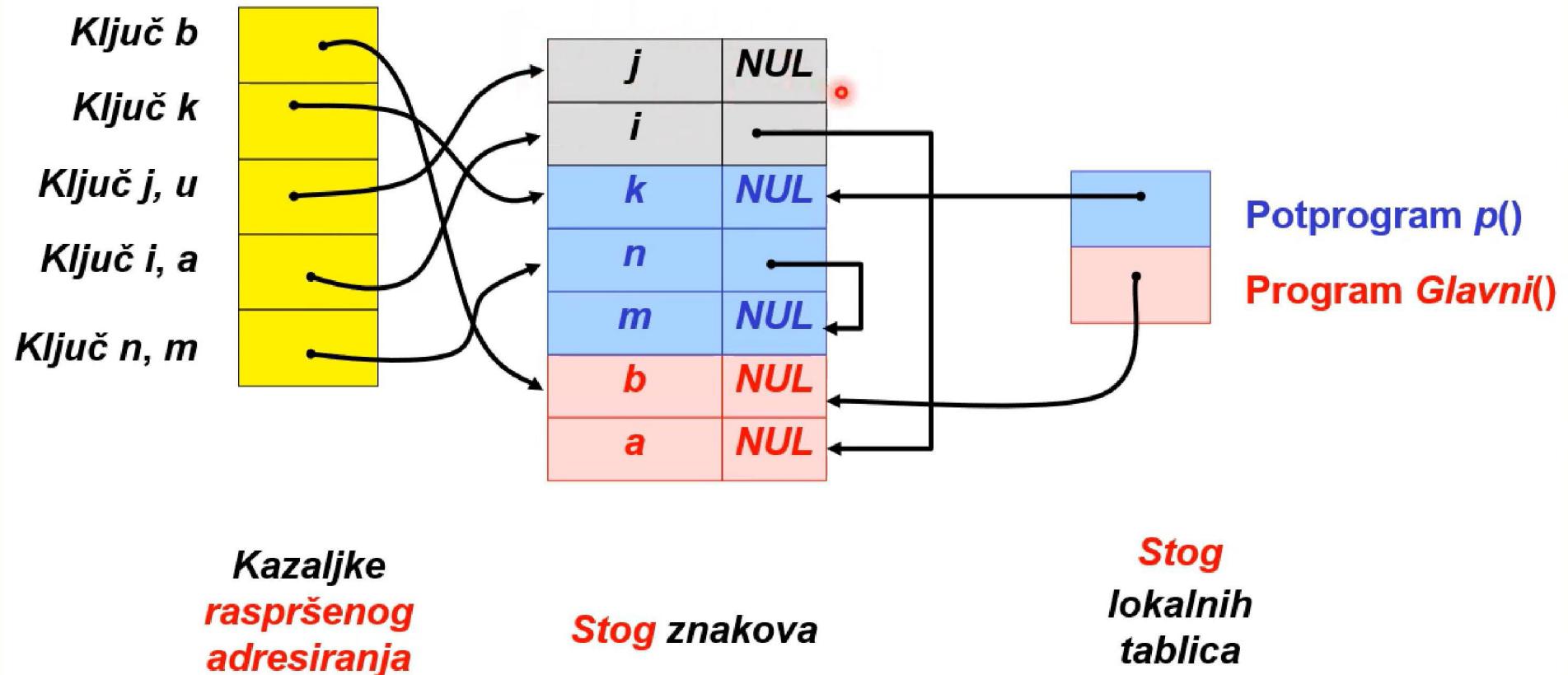
Programsko ostvarenje tablice znakova



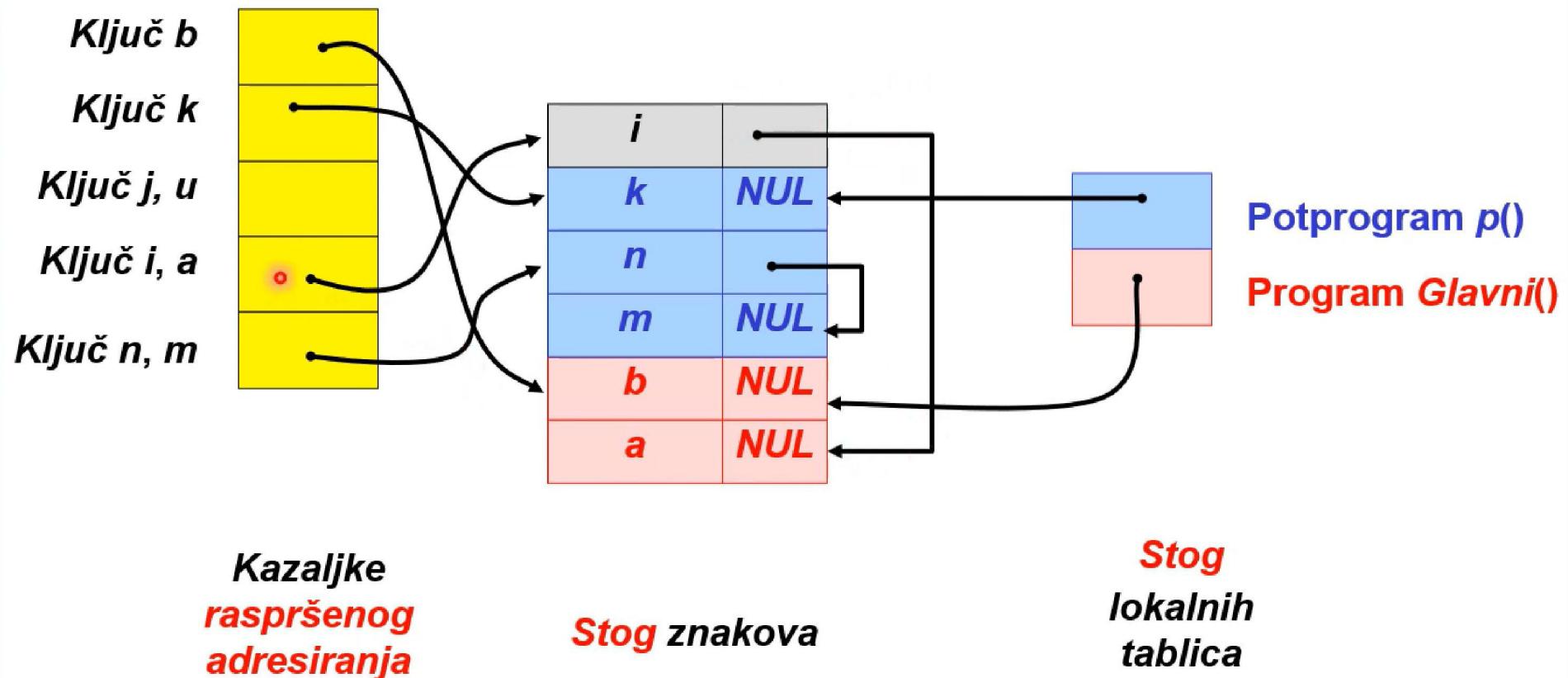
Programsko ostvarenje tablice znakova



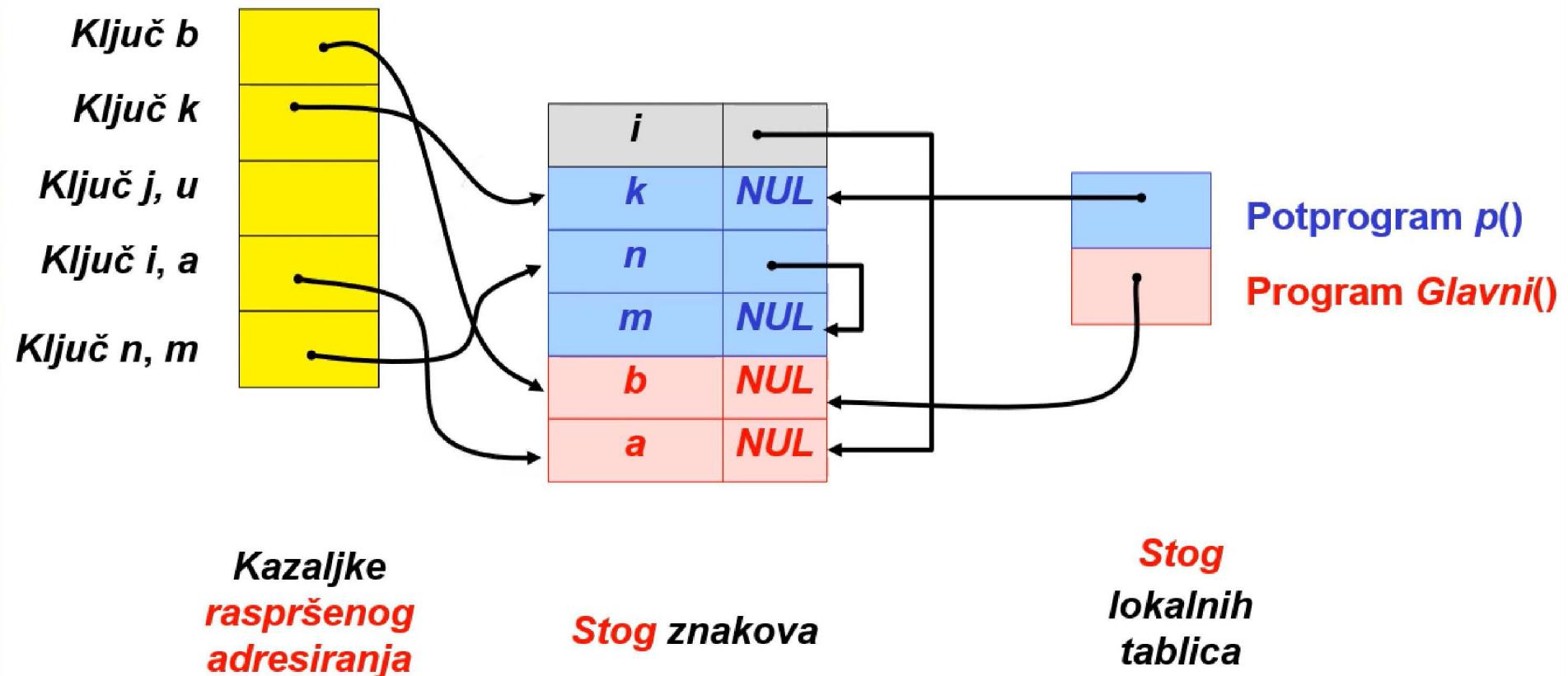
Programsko ostvarenje tablice znakova



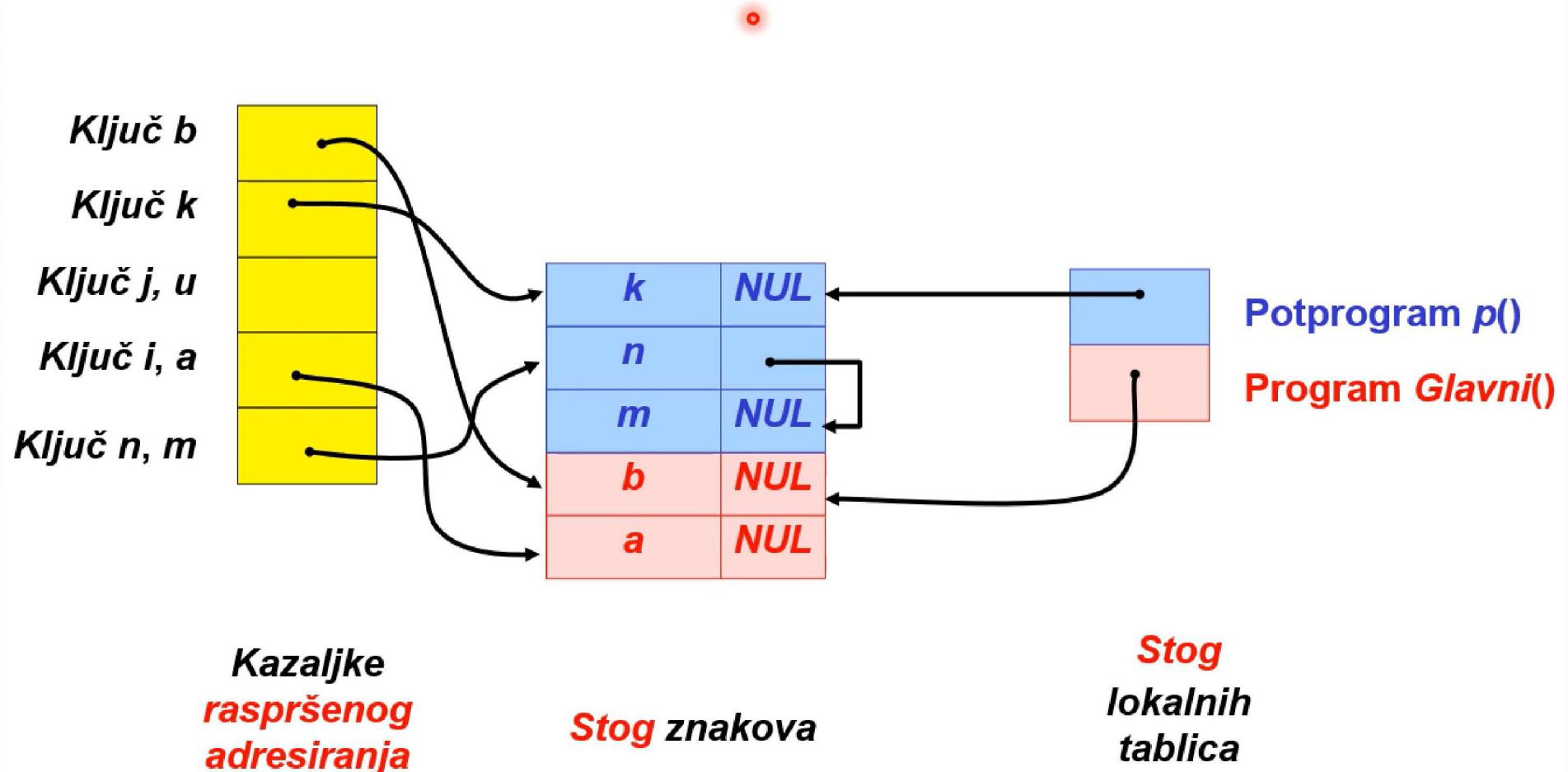
Programsko ostvarenje tablice znakova



Programsko ostvarenje tablice znakova



Programsko ostvarenje tablice znakova



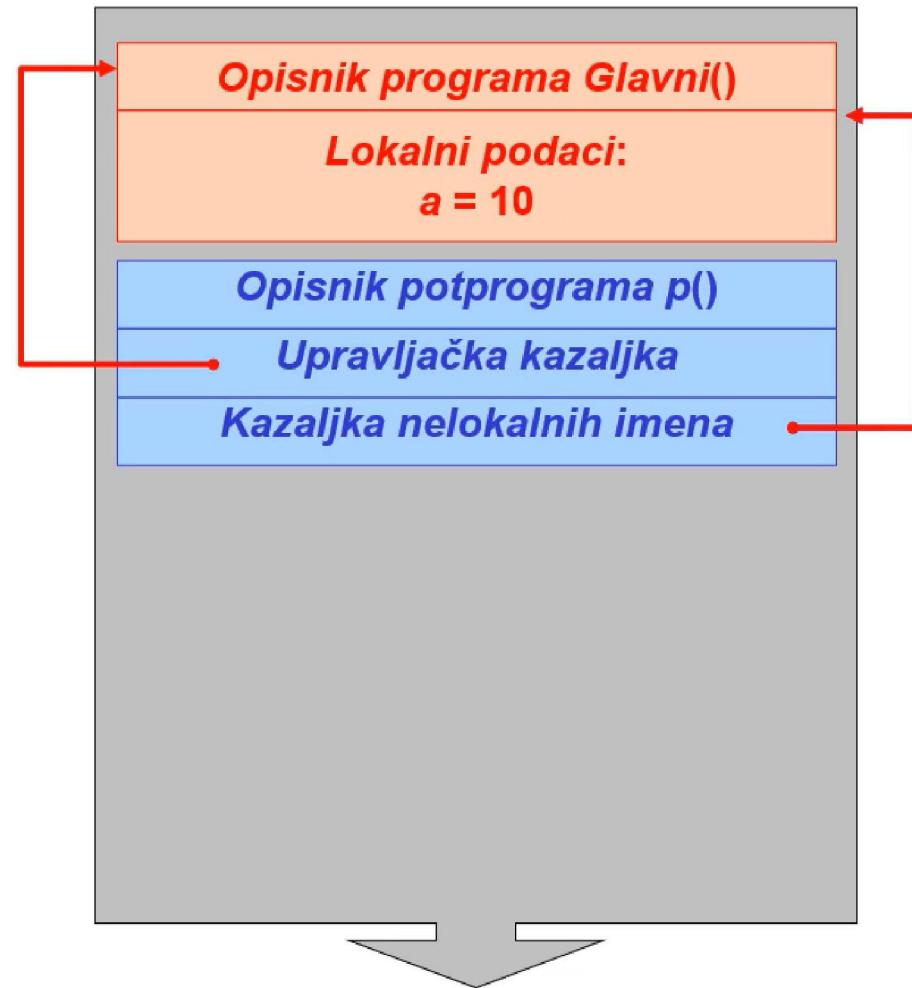
Dinamičko pravilo djelokruga

```
(1) Glavni()
(2) int a=10;

(3)     p()
(4) {
(5)     Ispisi(a);
(6) }

(7)     q()
(8)     int a=1;
(9) {
(10)    p();
(11) }

(12) {
(13)     p();
(14)     q();
(15) }
```



Dinamičko pravilo djelokruga

```
(1) Glavni()
(2) int a=10;

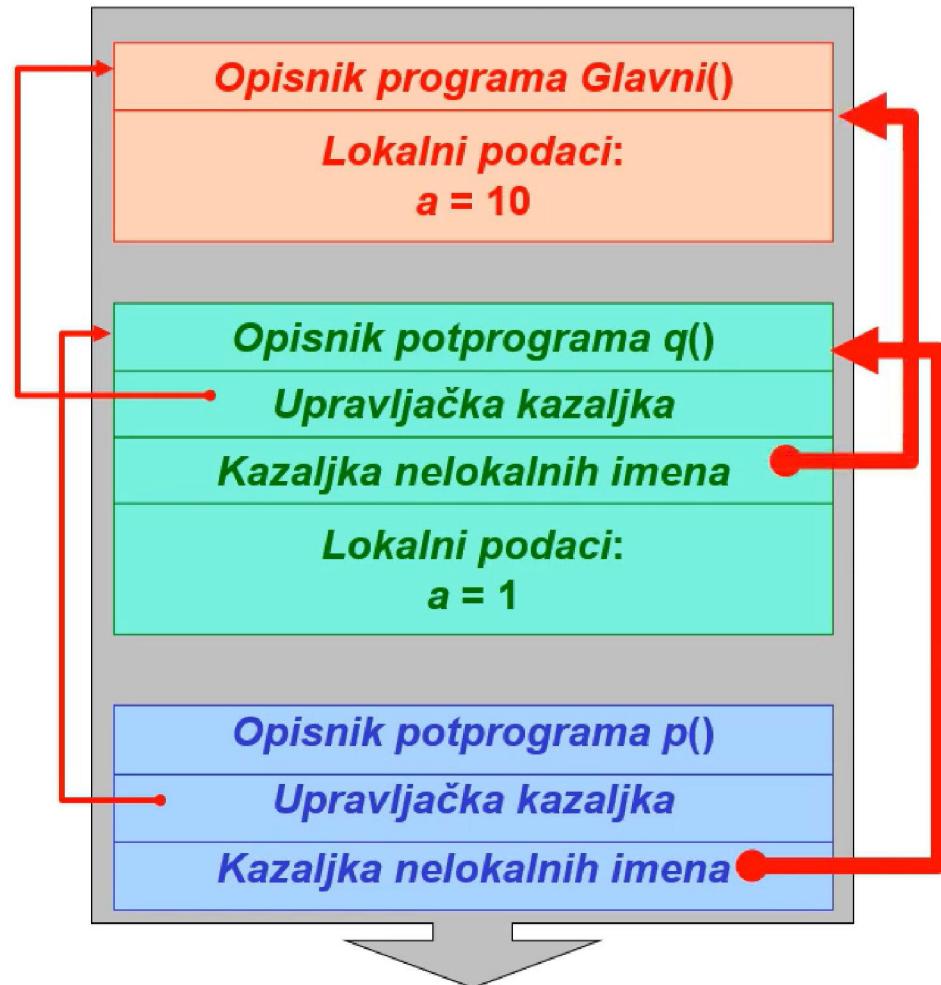
(3)     p()
(4) {
(5)     Ispisi(a);
(6) }

(7) q()
(8) int a=1;
(9) {
(10)    p();
(11) }

(12) {
(13)    p();
(14)    q();
(15) }
```

Diagram illustrating the dynamic scope rule (DPR) for variable 'a'. The code shows two scopes: 'Glavni()' (red border) and 'q()' (green border). In 'Glavni()', 'a' is initialized to 10. In 'q()', 'a' is redeclared and initialized to 1. The DPR allows both variables to be accessed. Red arrows indicate the flow of control from the main loop to the nested function 'q()' and back.

Pretraživanje po dubini



Dinamičko pravilo djelokruga

Spremanjem lokalno deklariranih vrijednosti

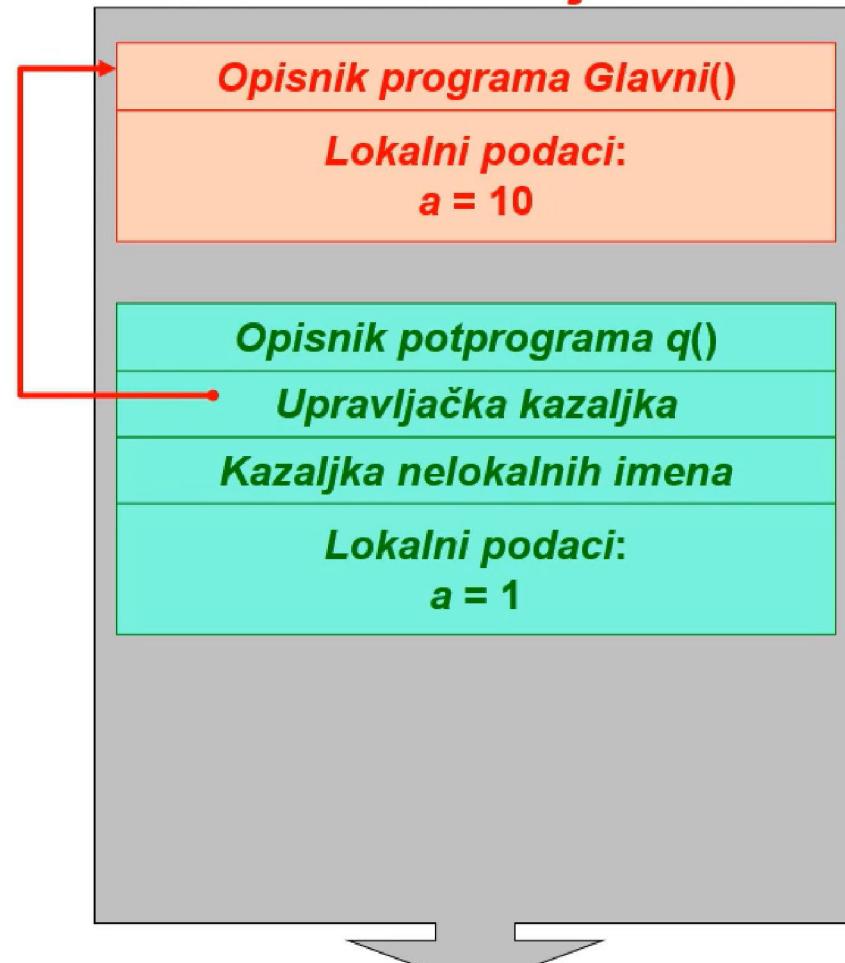
```
(1) Glavni()  
(2) int a=10;  
  
          p()  
(3)      {  
(4)          Ispisi(a);  
(5)      }  
  
          q()  
(6)      {  
(7)          q()  
(8)          int a=1;  
(9)          {  
(10)             p();  
(11)          }  
  
(12) {  
(13)     p();  
(14)     q();  
(15) }
```



Dinamičko pravilo djelokruga

Spremanjem lokalno
deklariranih vrijednosti

```
(1) Glavni()  
(2) int a=10;  
  
(3)     p()  
(4)     {  
(5)         Ispisi(a);  
(6)     }  
  
(7)     q()  
(8)     int a=1;  
(9)     {  
(10)        p();  
(11)    }  
  
(12) {  
(13)     p();  
(14)     q();  
(15) }
```



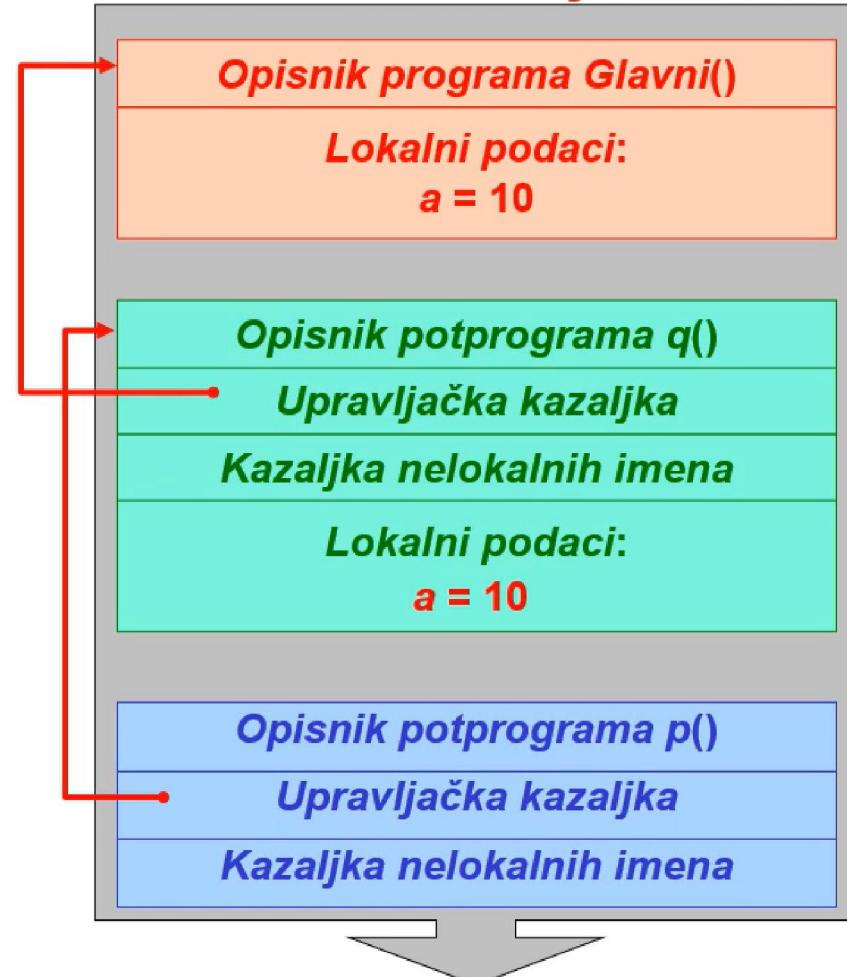
Statička memorija: a = 10



Dinamičko pravilo djelokruga

Spremanjem lokalno
deklariranih vrijednosti

```
(1) Glavni()  
(2) int a=10;  
  
(3)     p()  
(4)     {  
(5)         Ispisi(a);  
(6)     }  
  
(7) q()  
(8) int a=1;  
(9) {  
(10)    p();  
(11) }  
  
(12) {  
(13)     p();  
(14)     q();  
(15) }
```



Statička memorija: $a = 1$

Dinamičko pravilo djelokruga

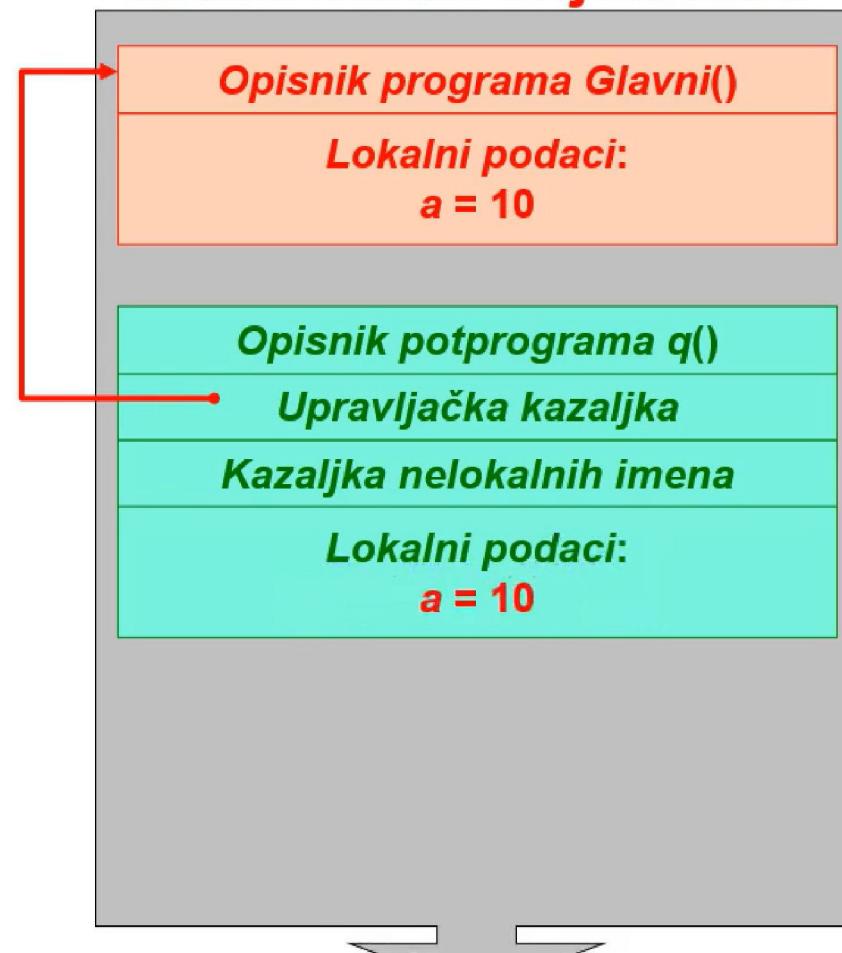
Spremanjem lokalno deklariranih vrijednosti

```
(1) Glavni()
(2) int a=10;

(3)     p()
(4) {
(5)     Ispisi(a);
(6) }

(7)     q()
(8)     int a=1;
(9) {
(10)    p();
(11) }

(12) {
(13)    p();
(14)    q();
(15) }
```



Statička memorija: a = 10



Dinamičko pravilo djelokruga

Spremanjem lokalno
deklariranih vrijednosti

```
(1) Glavni()  
(2) int a=10;  
  
(3)     p()  
(4)     {  
(5)         Ispisi(a);  
(6)     }  
  
(7)     q()  
(8)     int a=1;  
(9)     {  
(10)        p();  
(11)    }  
  
(12) {  
(13)     p();  
(14)     q();  
(15) }
```

Opisnik programa Glavni()

Lokalni podaci:
a = 10

Statička memorija: a = 10

