

Optimiranje

- **Mjera kvalitete ciljnog programa**
 - brzina izvođenja
 - veličina potrebnog memorijskog prostora
- **Mjera uspješnosti postupaka optimiranja**
 - Dosljednost
 - očuvanje značenja programa
 - agresivnost - skupljanje podataka o načinu izvođenja programa
 - pesimističnost - odlučivanje o načinu pretvorbe
 - Stupanj poboljšanja kvalitete ciljnog programa
 - Učinkovitost programa koji obavlja optimiranje

Optimiranje

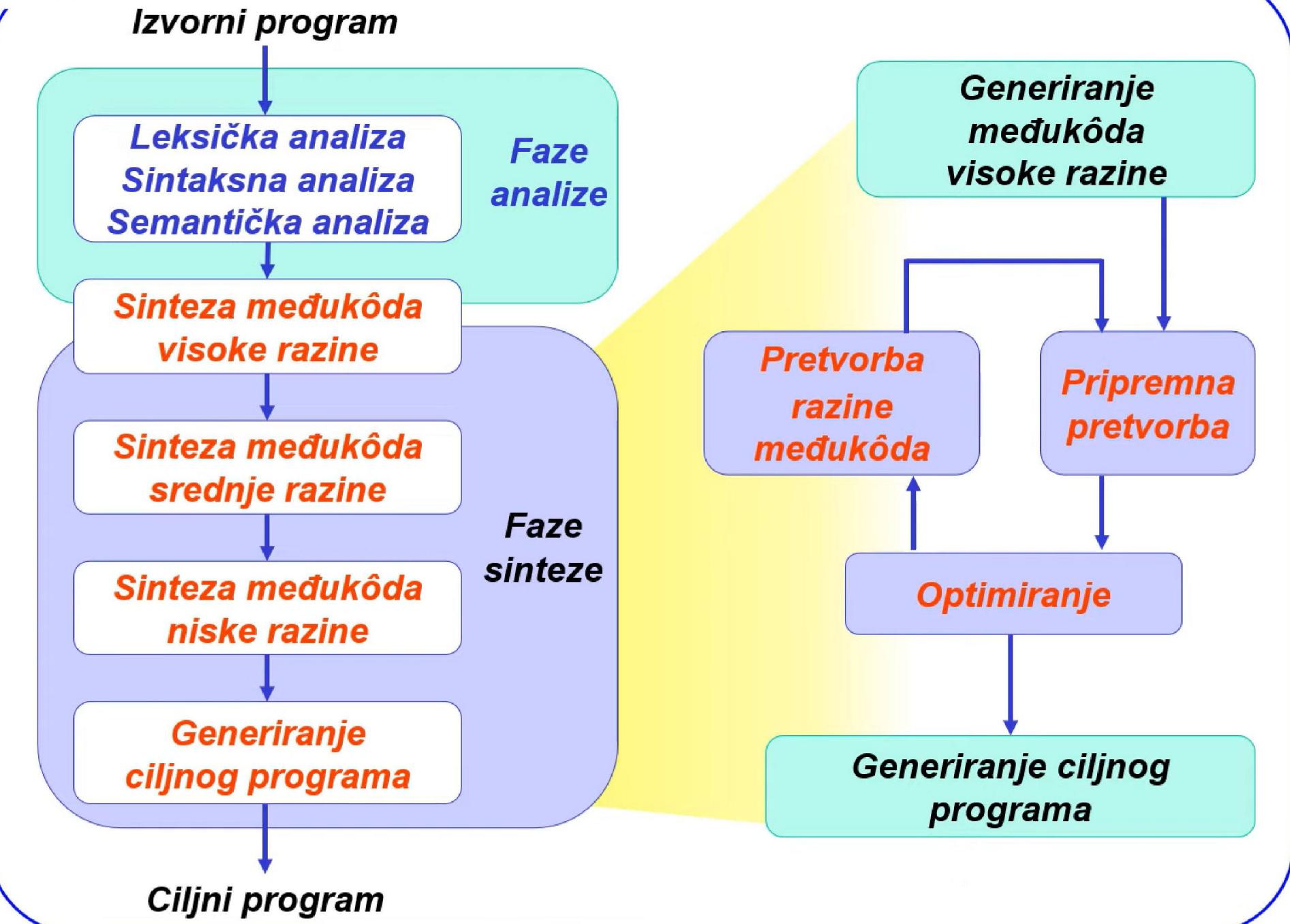
- **Učinkovitost optimiranja ovisi o**
 - **Programu koji se optimira**
 - programer najviše doprinosi kvaliteti ciljnog programa učinkovitim programskim ostvarenjem pravilno izabranog algoritma
 - **Dostupnosti informacije o namjeni izvornog programa**
 - područjem primjene određen je relativni odnos radnih svojstava programa
 - brzina izvođenja, potrošnja memorije, potrošnja energije
 - **Mogućnost određivanja koje se naredbe programa izvode češće od ostalih naredbi**
 - vrijeme izvođenja 1% naredbi programa čini 99% vremena izvođenja cijelokupnog programa

Optimiranje

- **Složen postupak**
 - **Više jednostavnijih postupaka postupne sinteze ciljnog programa**
 - **Dvorazinski postupak**
 - strojno nezavisni postupci
 - optimiraju programske strukture bliske strukturama višeg programskog jezika
 - » programske petlje, složene naredbe grananja, procedure, pristup složenim apstraktnim tipovima podataka
 - strojno zavisni postupci
 - optimiraju programske strukture bliske strukturama strojnog jezika
 - postiže se učinkovita uporaba raznih sredstava računala
 - » memorijske hijerarhije, naredbenog skupa procesora, procesorskog cjevovoda

Optimiranje

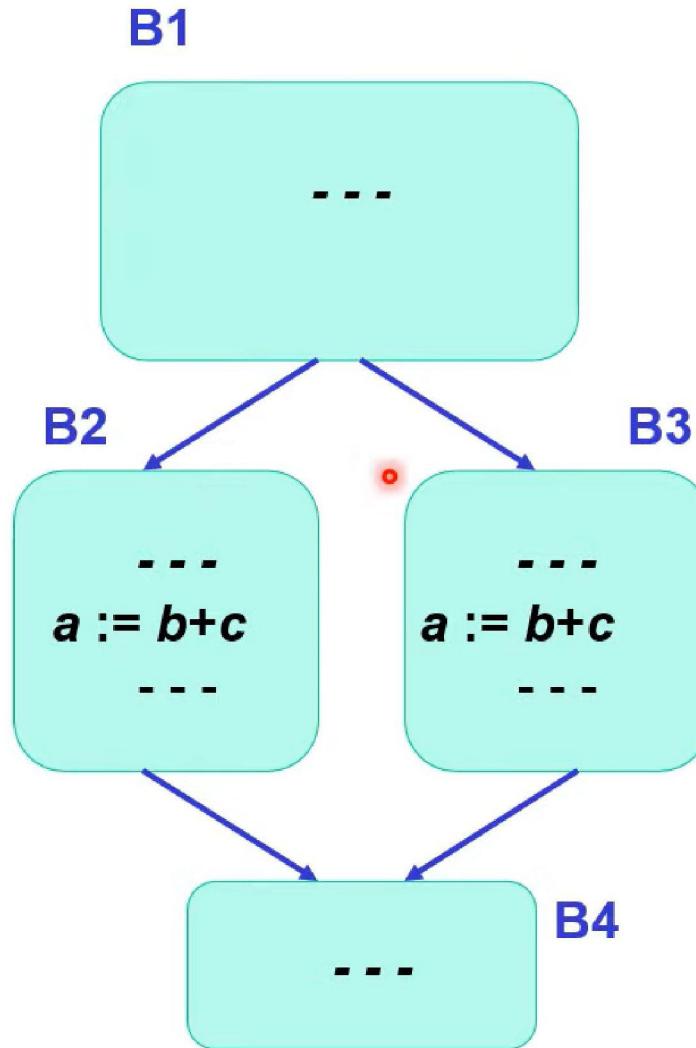
- **Algoritmi optimiranja**
 - **NP-potpuni ili nema optimalnog rješenja**
 - **Postupci pretvorbe**
 - postupno poboljšanje kvalitete ciljnog programa
 - **Lokalni postupci**
 - ispituju se naredbe jednog osnovnog bloka
 - **Globalni postupci**
 - ispituje se više osnovnih blokova
 - **Unutarproceduralni postupci**
 - ispituju naredbe isključivo jedne procedure
 - **Međuproceduralni postupci**
 - ispituju naredbe više procedura



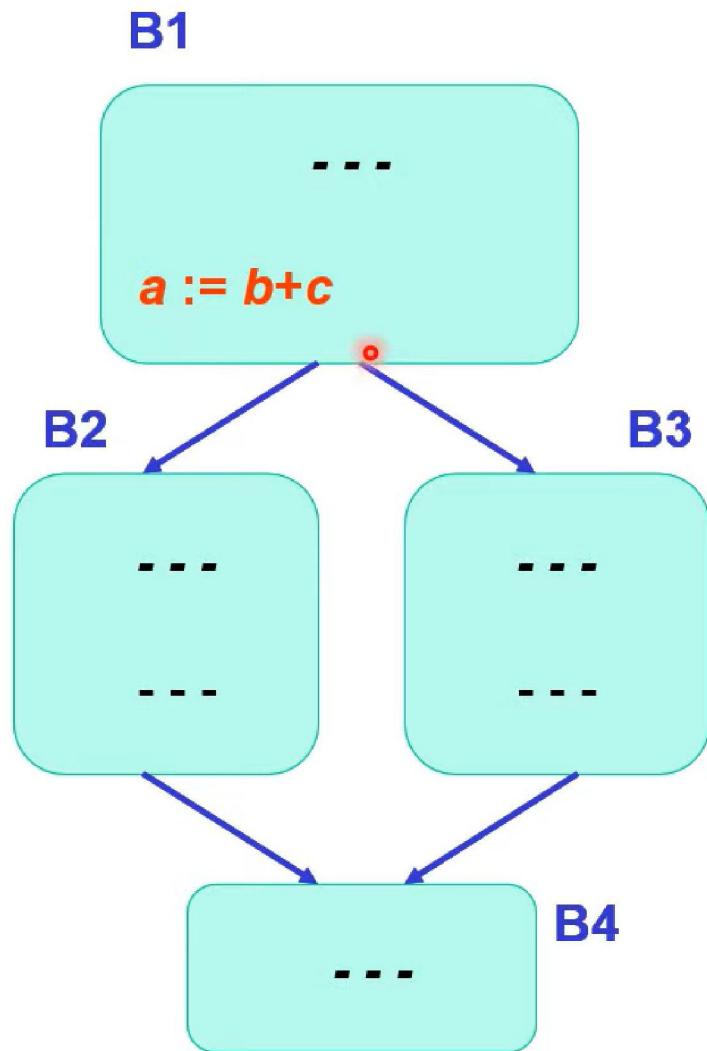
Sinteza ciljnog programa

- **Primjeri postupaka za optimiranje izraza**
 - **Između dva računanja istog izraza vrijednosti njegovih operanada se ne mijenjaju** •
 - iz međukoda se izuzmu one naredbe koje računaju taj izraz po drugi put
 - **Naredbe petlje ne mijenjaju vrijednosti operanda izraza**
 - naredba koja računa taj izraz izluči se izvan petlje

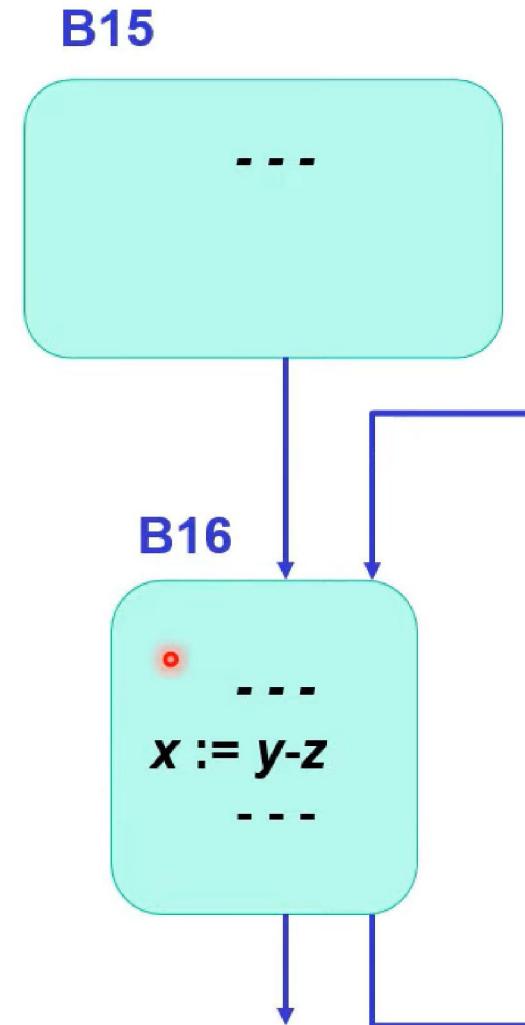
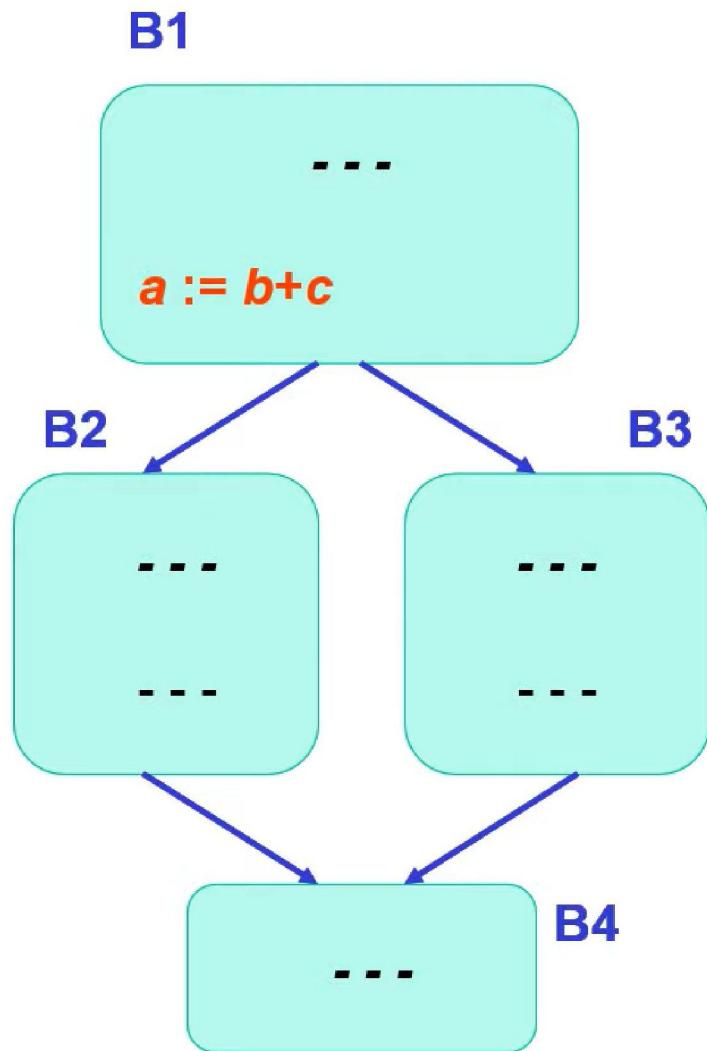
Optimiranje izraza



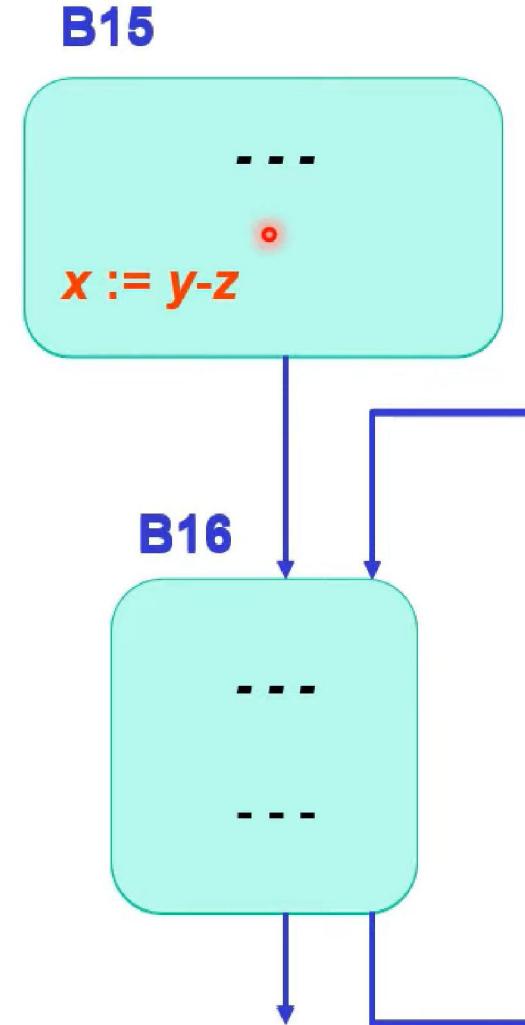
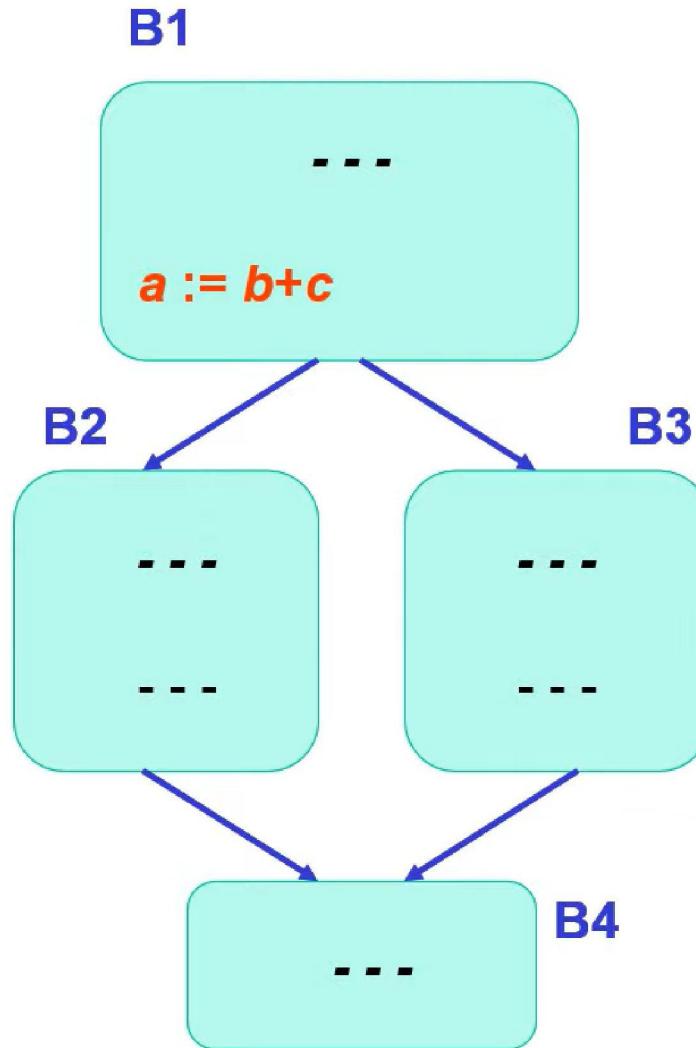
Optimiranje izraza



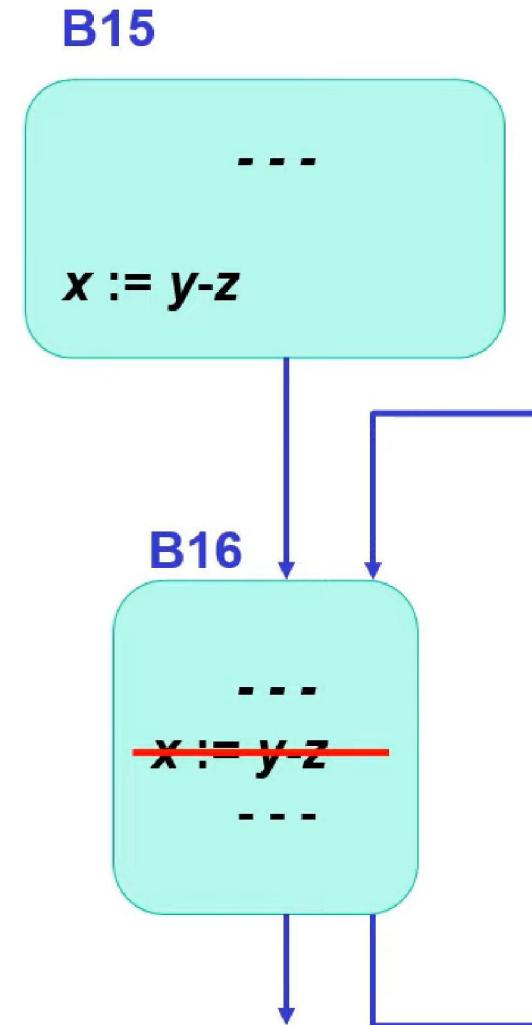
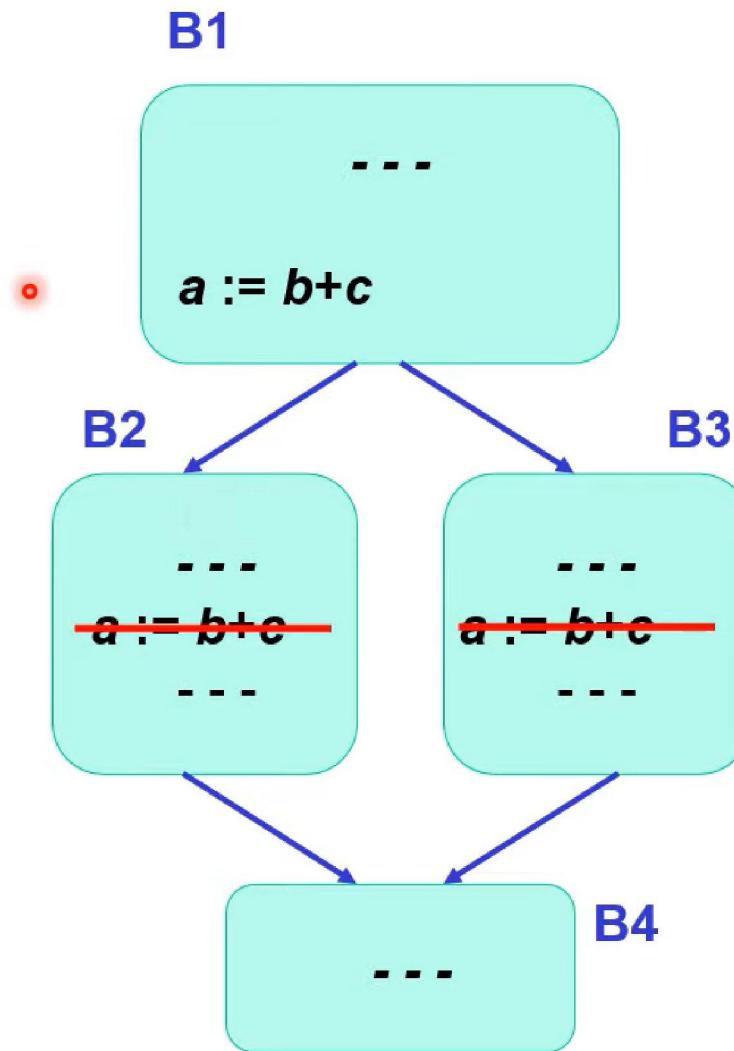
Optimiranje izraza



Optimiranje izraza

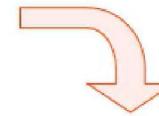


Postupak optimiranja

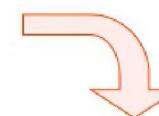


Analiza izvođenja programa

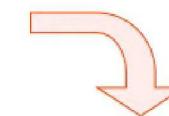
Analiza tijeka izvođenja programa



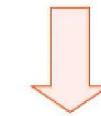
Analiza toka podataka



Analiza zavisnosti podataka



Analiza pseudonima



Postupci pretvorbe

Analiza tijeka izvođenja programa

- Ispituje upravljačku strukturu međukôda definiranu naredbama petlji, grananja, itd.
- Dva osnovna pristupa:
 - **Analiza dominacije**
 - **Analiza strukture**
- Oba pristupa grade:
 - **Graf tijeka izvođenja programa**
- Analiza dominacije
 - određuje strukturu petlji
 - prethodi analizi toka podataka zasnovanoj na *iterativnim postupcima*
- Analiza strukture
 - određuje više različitih podgrafova u grafu tijeka izvođenja programa
 - prethodi analizi toka podataka zasnovanoj na *eliminacijskim postupcima*

Graf tijeka izvođenja programa

- **Osnovni blokovi**
 - **niz naredbi koje se slijedno izvode**
 - **izvođenje osnovnog bloka**
 - započinje isključivo početnom naredbom bloka
 - osim završne naredbe bloka, nijedna druga naredba unutar bloka ne upravlja tijekom izvođenja programa
- **Graf tijeka izvođenja programa**
 - **Čvorovi grafa**
 - **osnovni blokovi**
 - **Usmjerene grane**
 - **definiraju tijek izvođenja programa**
 - **neposredno nakon izvedene završne naredbe jednog bloka**
slijedi izvođenje početne naredbe drugog bloka

Graf tijeka izvođenja programa

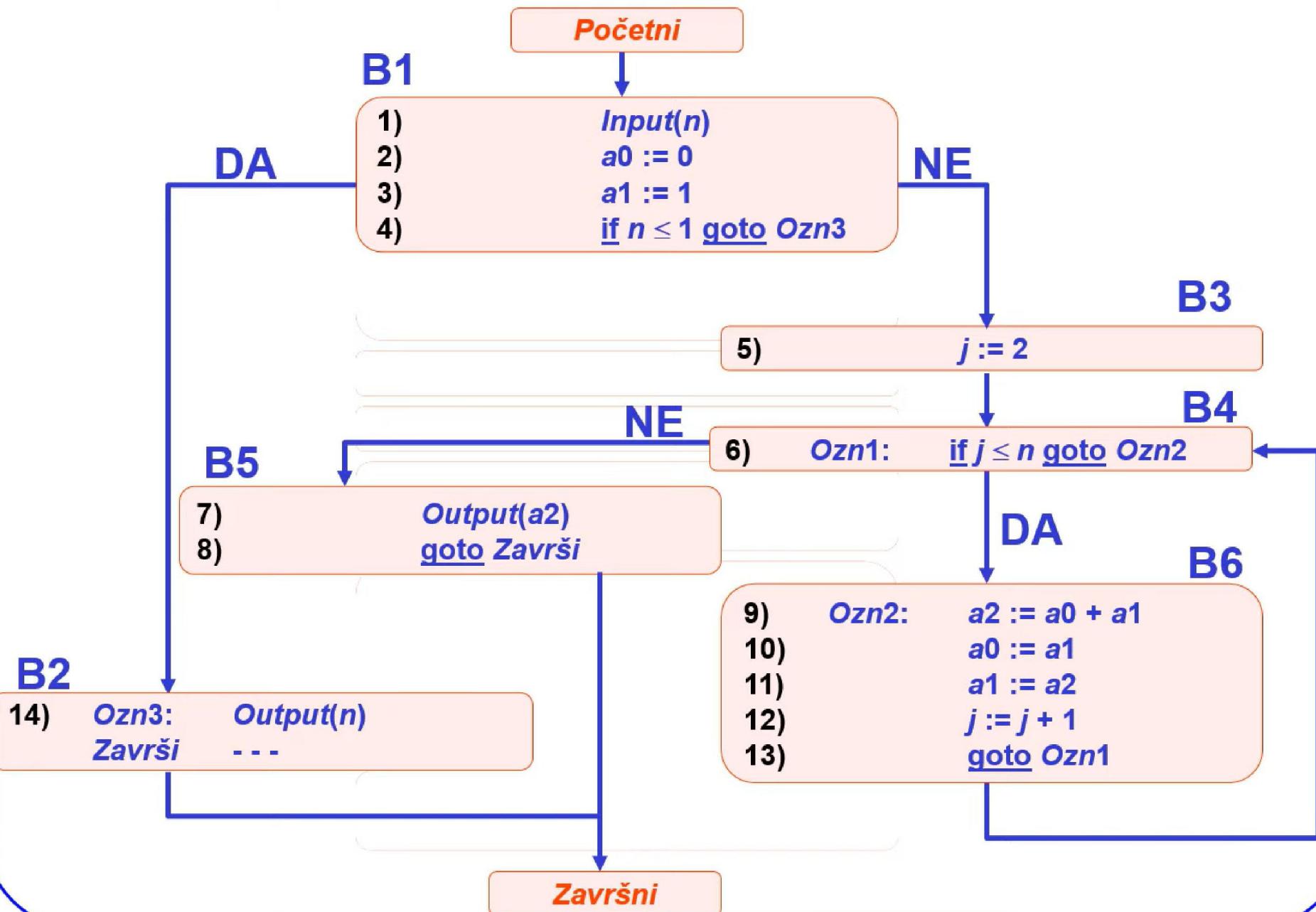
Početni



- 1) *Input(n)*
- 2) *a0 := 0*
- 3) *a1 := 1*
- 4) *if n ≤ 1 goto Ozn3*
- 5) *j := 2*
- 6) *Ozn1: if j ≤ n goto Ozn2*
- 7) *Output(a2)*
- 8) *goto Završi*
- 9) *Ozn2: a2 := a0 + a1*
- 10) *a0 := a1*
- 11) *a1 := a2*
- 12) *j := j + 1*
- 13) *goto Ozn1*
- 14) *Ozn3: Output(n)*
- Završi: ...

Završni

Graf tijek izvođenja programa



Graf tijeka izvođenja programa

- Početne naredbe blokova su
 - (1) početna naredba programa
 - (2) naredbe na koje se usmjeruje tijek izvođenja programa
 - (3) naredbe koje se nalaze neposredno iza naredbe grananja
- Osnovni blok čini početna naredba i sve naredbe između te i sljedeće početne naredbe
- Blokovi *A* i *B* povezuju se usmjerrenom granom ako i samo ako je istinito:
 - (1) završna naredba u bloku *A* usmjeruje tijek izvođenja programa na početnu naredbu bloka *B*
 - (2) blok *B* je neposredno iza bloka *A* koji ne završava bezuvjetnim preusmjeravanjem tijeka izvođenja programa

Analiza dominacije

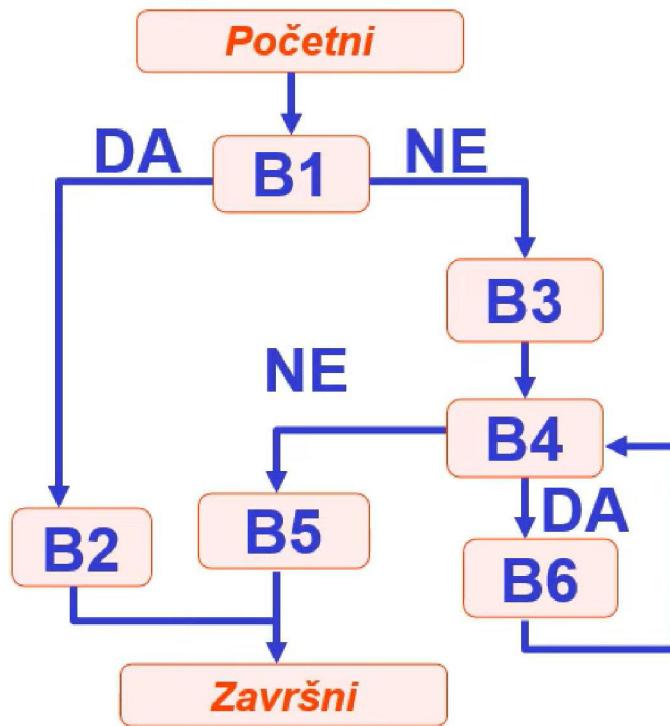
- **Za potrebe postupaka optimiranja**
 - Potrebno je odrediti strukturu programskih petlji
 - izlučivanje izraza iz petlji
 - optimiranje spregnutih varijabli
 - istodobnost izvođenja dijelova petlje
 - Petlja
 - podgraf tijeka izvođenja programa
 - određuje se na temelju relacije dominacije
- **Čvor d dominira čvorom a**
 - čvor d prethodi čvoru a na svakom putu koji započinje početnim čvorom *Početni* i prolazi čvorom a
 - $\text{dom}(a)$
 - skup koji sadrži sve dominatore čvora a

Analiza dominacije

- Blok je sam svoj dominator
- Čvor d striktno dominira čvorom a
 - $d \in \text{dom}(a)$ i $d \neq a$
- Čvor d je neposredni dominator čvora a
 - ni za jedan drugi čvor c ($c \neq d$ i $c \neq a$) ne vrijedi
 - $d \in \text{dom}(c)$
 - $c \in \text{dom}(a)$
- Stablo dominacije
 - čvor d je neposredni dominator čvora a
 - u stablu dominacije čvor d jest neposredni prethodnik (roditelj) čvora a
- Čvor p postdominira čvorom a
 - čvor p slijedi čvor a na svakom putu koji započinje čvorom a i završava završnim čvorom **Završni**

Analiza dominacije

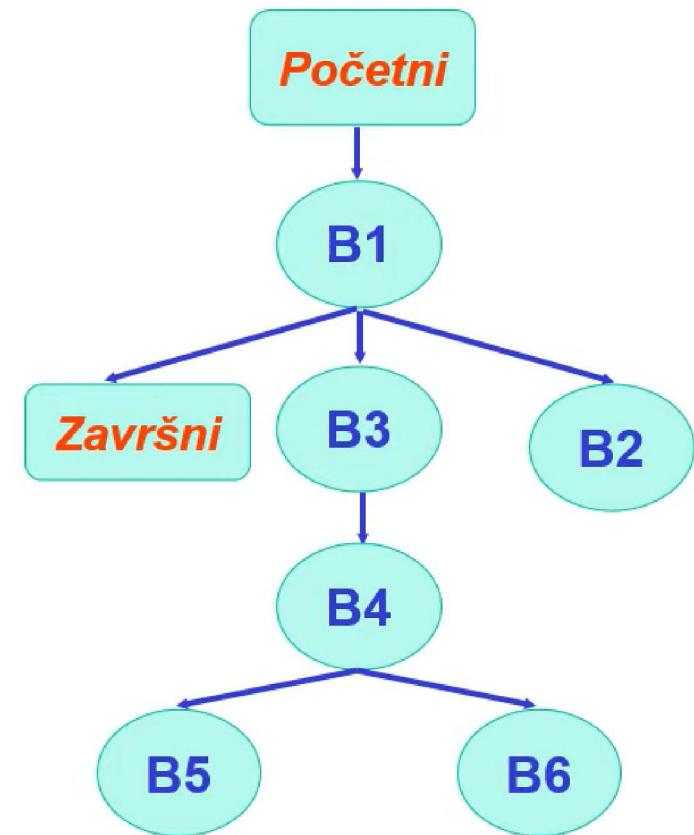
Graf tijeka izvođenja programa



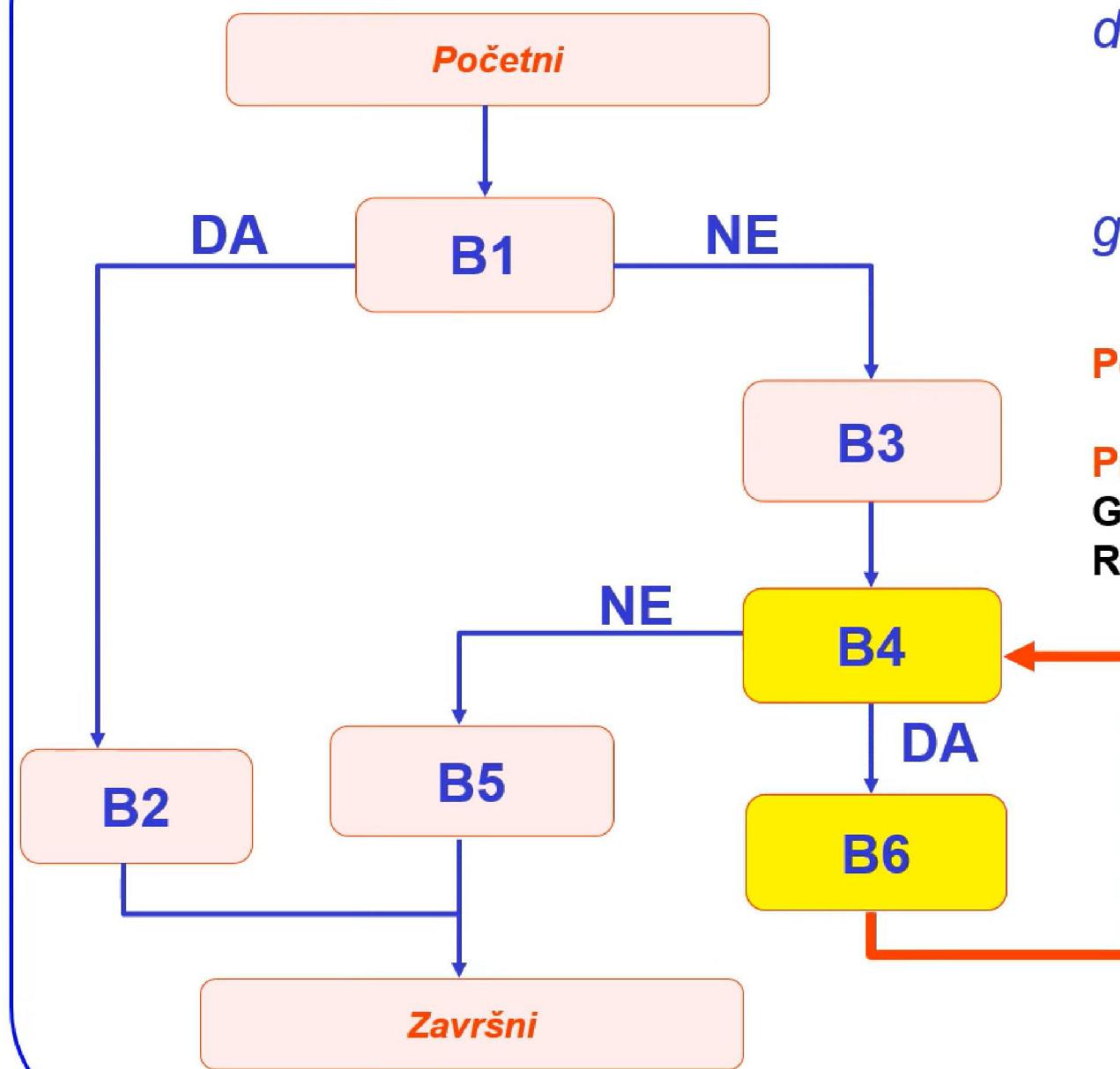
Relacija dominacije

$dom(Početni)$	=	{Početni}
$dom(B1)$	=	{Početni, B1}
$dom(B2)$	=	{Početni, B1, B2}
$dom(B3)$	=	{Početni, B1, B3}
$dom(B4)$	=	{Početni, B1, B3, B4}
$dom(B5)$	=	{Početni, B1, B3, B4, B5}
$dom(B6)$	=	{Početni, B1, B3, B4, B6}
$dom(Završni)$	=	{Početni, B1, Završni}

Stablo dominacije



Analiza dominacije



$dom(B6) = \{Početni, B1, B3, B4, B6\}$

grana = (B6, B4)

Povratna grana B6 → B4

Prirodna petlja

Glava petlje

Rep petlje

B4

B6

- **Čvorovi prirodne petlje**
 - glava petlje
 - svi čvorovi za koje postoji put do repa petlje, uz uvjet da na putu nije glava petlje
 - primjer prirodne petlje
 - skup čvorova {B4, B6}
 - skup usmjerenih grana {B6→B4, B4→B6}
- **Skup čvorova petlje**
 - rep petlje
 - svi neposredni prethodnici repa koji nisu prethodnici glave
 - svi neposredni prethodnici čvorova u skupu, uz uvjet da oni nisu prethodnici glave

- **Nedostatak analize dominacije**
 - nemogućnost njezinog učinkovitog ponavljanja
 - postupci optimiranja mijenjaju strukturu programa
 - želi li se primijeniti sljedeći postupak optimiranja
 - potrebno je ponoviti cjelokupnu analizu dominacije zajedno s iterativnim postupkom analize toka podataka
 - vremenski zahtjevno

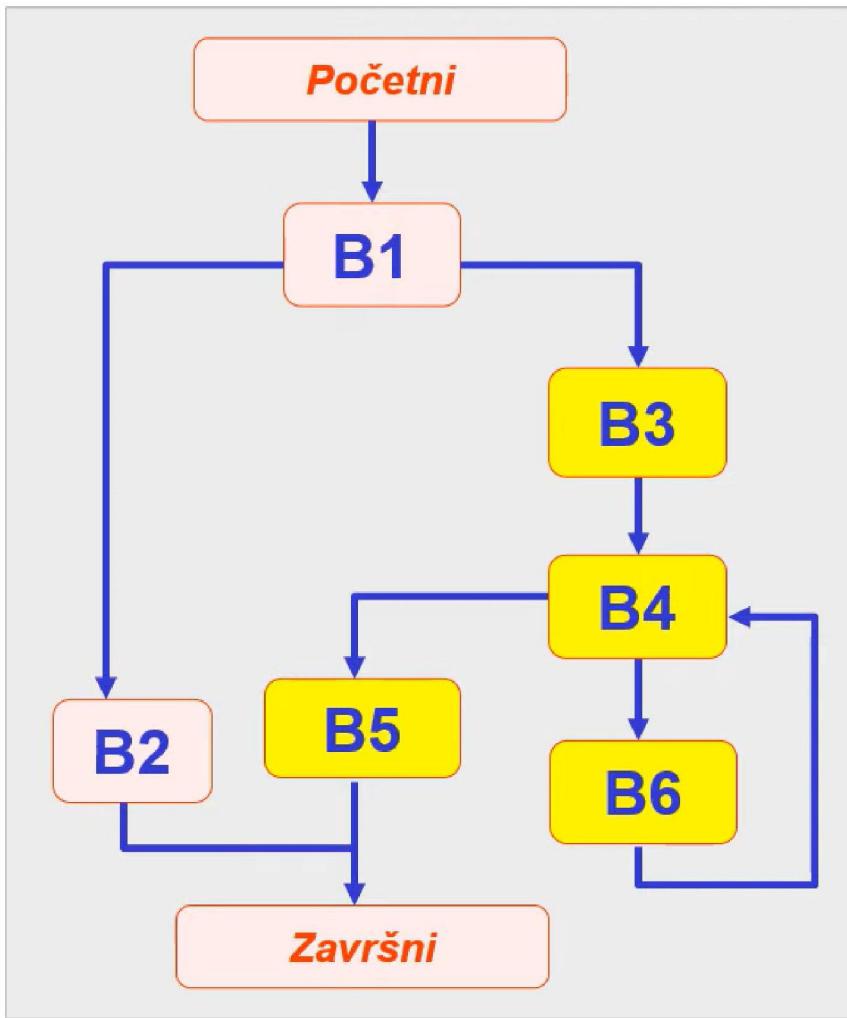
Analiza strukture

- **Analiza strukture gradi**
 - upravljačko stablo
 - na temelju raščlambe grafa tijeka izvođenja programa
- **Upravljačko stablo**
 - hijerarhijski zapis strukture grafa tijeka izvođenja programa
 - podgrafovi uzorci
 - slijedni čvorovi, čvor s petljom, if-then-else čvorovi, itd.
- **Analiza strukture**
 1. obilazi graf tijeka izvođenja programa
 2. traži podgrafove uzorke
 3. zamjeni ih jednim zamjenskim čvorom
 4. gradi upravljačko stablo

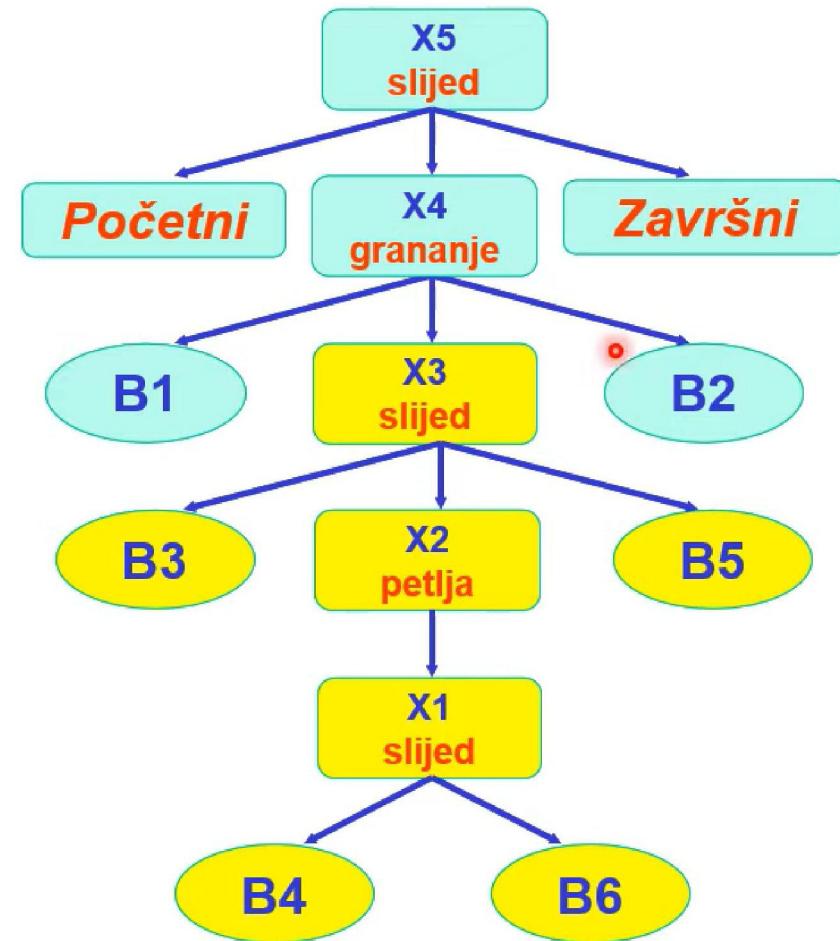
- **Gradnja upravljačkog stabla**
 - **započinje listovima**
 - **listovi upravljačkog stabla**
 - čvorovi grafa tijeka izvođenja programa
 - **Zamjenom podgraфа uzorka zamjenskim čvorom**
 - smanji se broj čvorova grafa tijeka izvođenja programa
 - u upravljačko stablo doda se zamjenski čvor
 - zamjenski čvor u upravljačkom stablu je prethodnik čvorova podgraфа uzorka

Analiza strukture

Raščlamba grafa tijekom izvođenja programa



Gradnja upravljačkog stabla



- **Izravni nastavak analize tijeka izvođenja programa**
 - Iterativni postupak
 - koristi analizu dominacije grafa tijeka izvođenja programa
 - Eliminacijski postupak
 - koristi analizu strukture
- **Obilazak grafa tijeka izvođenja programa**
 - Unaprijedna analiza
 - u smjeru izvođenja programa
 - Unazadna analiza
 - u smjeru suprotnom od smjera izvođenja programa
 - Dvosmjerna analiza
 - u oba smjera

Analiza toka podataka

- **Analiza**

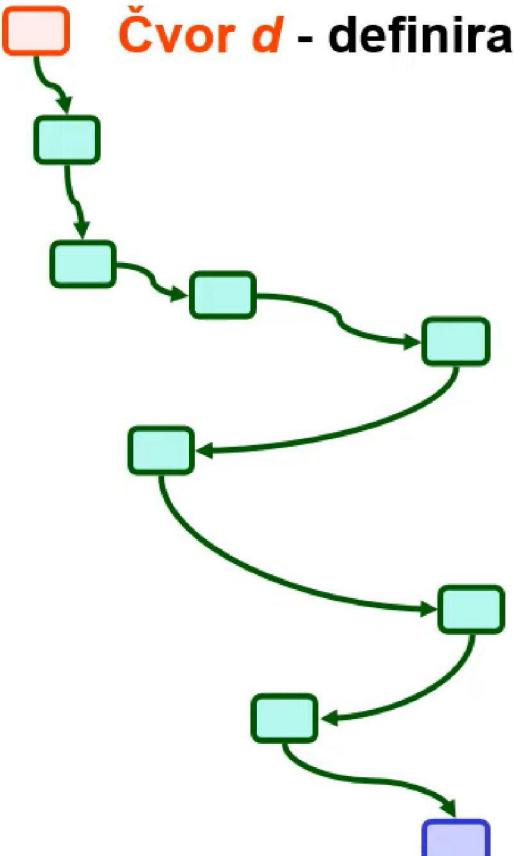
- dosega definicije varijable
- važećih definicija varijable
- dostupnih izraza
- živih varijabli
- prenošenja naredbi preslikavanja
- prenošenje konstanti
- višekratnog računanja izraza

Doseg definicije varijable

- **Unaprijedna analiza**
- **Skup čvorova koji koriste definiciju varijable**
- **Zapisuje se bit vektorom**
- **Vrijednost bita 1**
 - ulaz ili izlaz čvora je u dosegu definicije varijable

Doseg definicije varijable

 Čvor **d** - definira vrijednost varijable



Nijedan čvor puta
nanovo ne definira
vrijednost varijable

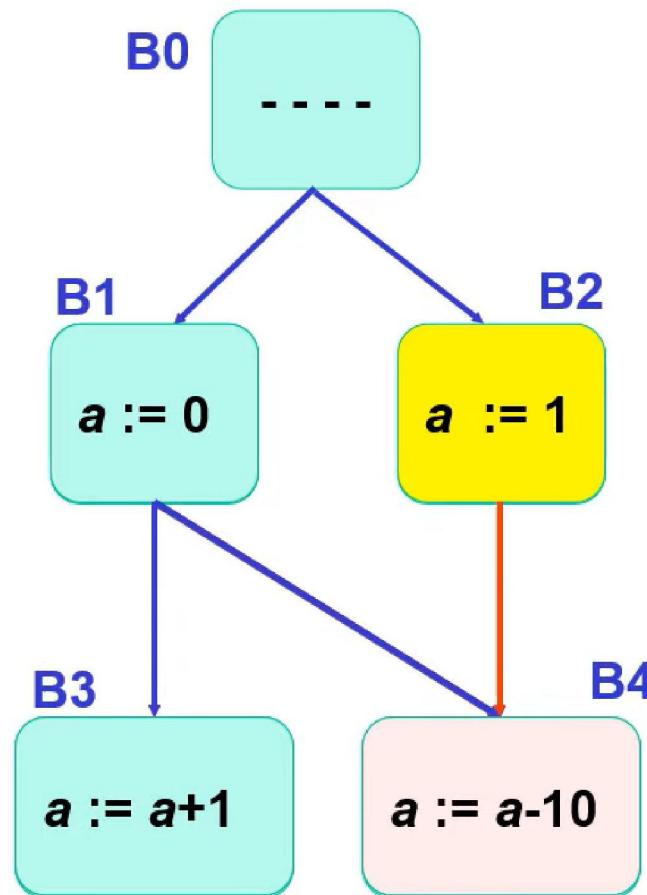
 Čvor **u** – koristi vrijednost varijable

Čvor **u** je u dosegu definicije čvora **d**

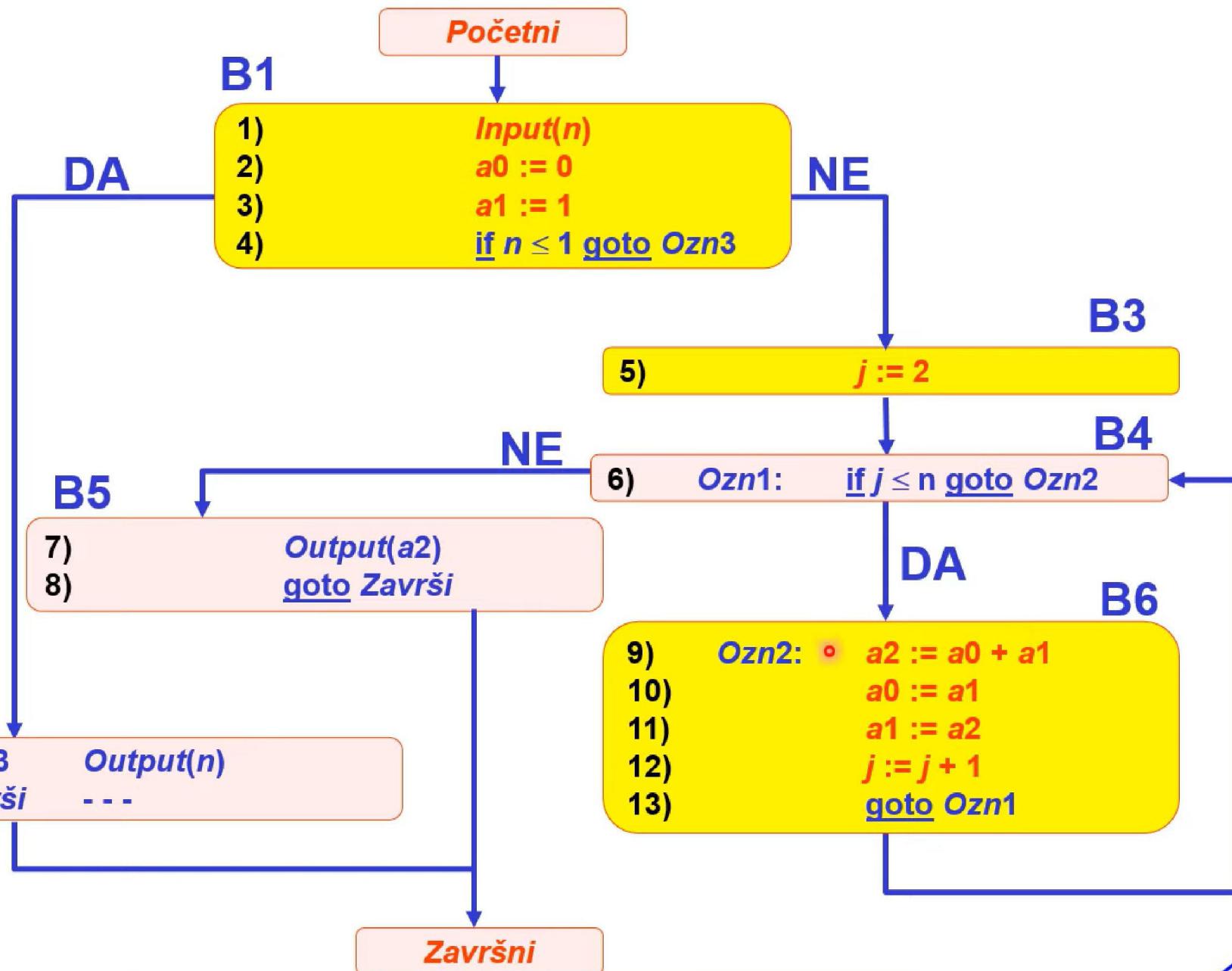
Doseg definicije varijable

U dosegu definicije bloka **B1** su blokovi **{B3, B4}**

U dosegu definicije bloka **B2** je blok **{B4}**



Iterativni postupak računanja dosega definicije varijable



Iterativni postupak računanja dosega definicije varijable

<i>Mjesto bita</i>	<i>Definicija</i>	<i>Osnovni blok</i>
1	<i>n</i> u liniji (1)	B1
2	<i>a0</i> u liniji (2)	B1
3	<i>a1</i> u liniji (3)	B1
4	<i>j</i> u liniji (5)	B3
5	<i>a2</i> u liniji (9)	B6
6	<i>a0</i> u liniji (10)	B6
7	<i>a1</i> u liniji (11)	B6
8	<i>j</i> u liniji (12)	B6

Gen(B1) = <11100000>

Gen(B3) = <00010000>

Gen(B6) = <00001111>

Gen(n) = <00000000>

Čuva(B1) = <00011001>

Čuva(B3) = <11101110>

Čuva(B6) = <10000000>

Čuva(n) = <11111111>

Jednadžbe toka podataka

$$\text{Izlaz}(a) = \text{Gen}(a) \vee (\text{Ulaz}(a) \wedge \text{Čuva}(a))$$

$$\text{Ulaz}(a) = \bigvee_i \text{Izlaz}(i)$$

Početne vrijednosti

$$\text{Izlaz}(a) = \text{Ulaz}(a) = <0000\dots 0>$$

•

$U\text{laz}(B3) = I\text{zlaz}(B1) = <11100000>$

$$\begin{aligned} I\text{zlaz}(B3) &= G\text{en}(B3) \quad \vee (U\text{laz}(B3) \quad \wedge \check{C}\text{uva}(B3)) = \\ &<00010000> \quad \vee (<11100000> \quad \wedge <11101110>) = \\ &<11110000> \end{aligned}$$

$U\text{laz}(Početni) = <00000000>$

$U\text{laz}(B1) = <00000000>$

$U\text{laz}(B2) = <11100000>$

$U\text{laz}(B3) = <11100000>$

$U\text{laz}(B4) = <11110000>$

$U\text{laz}(B5) = <11110000>$

$U\text{laz}(B6) = <11110000>$

$U\text{laz}(Završni) = <11110000>$

$I\text{zlaz}(Početni) = <00000000>$

$I\text{zlaz}(B1) = <11100000>$

$I\text{zlaz}(B2) = <11100000>$

$I\text{zlaz}(B3) = <11110000>$

$I\text{zlaz}(B4) = <11110000>$

$I\text{zlaz}(B5) = <11110000>$

$I\text{zlaz}(B6) = <10001111>$

$I\text{zlaz}(Završni) = <11110000>$

Ulaz(Početni) = <00000000>

Ulaz(B1) = <00000000>

Ulaz(B2) = <11100000>

Ulaz(B3) = <11100000>

Ulaz(B4) = <11111111>

Ulaz(B5) = <11111111>

Ulaz(B6) = <11111111>

Ulaz(Završni) = <11111111>

Izlaz(Početni) = <00000000>

Izlaz(B1) = <11000000>

Izlaz(B2) = <11100000>

Izlaz(B3) = <11110000>

Izlaz(B4) = <11111111>

Izlaz(B5) = <11111111>

Izlaz(B6) = <10001111>

Izlaz(Završni) = <11111111>

Definicija varijable **a0** u bloku B1



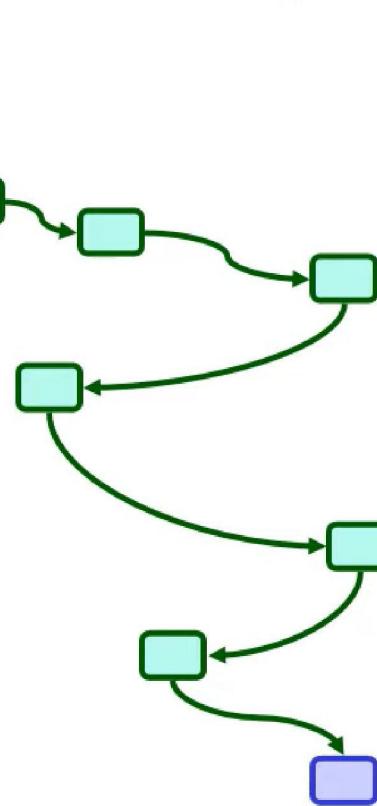
doseže do bloka **B6**

Važeće definicije varijable

- **Unazadna analiza**
- **Skup čvorova koji definiraju vrijednost varijable**
- **Bit vektor**
 - zasebni bitovi dodjeljuju se različitim uporabama varijable

Važeće definicije varijable

 Čvor **d** - definira vrijednost varijable



Nijedan čvor puta
nanovo ne definira
vrijednost varijable

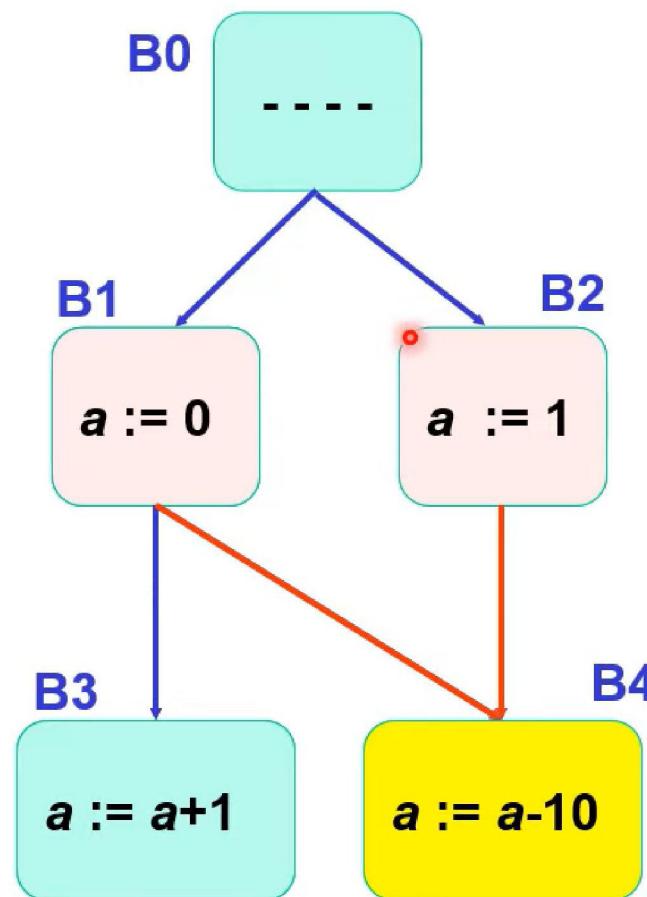
 Čvor **u** – koristi vrijednost varijable

Definicija čvora **d** je važeća za čvor **u**

Važeće definicije varijable

Važeća definicija za uporabu varijable **a** u bloku **B3** je definicija iz bloka **{B1}**

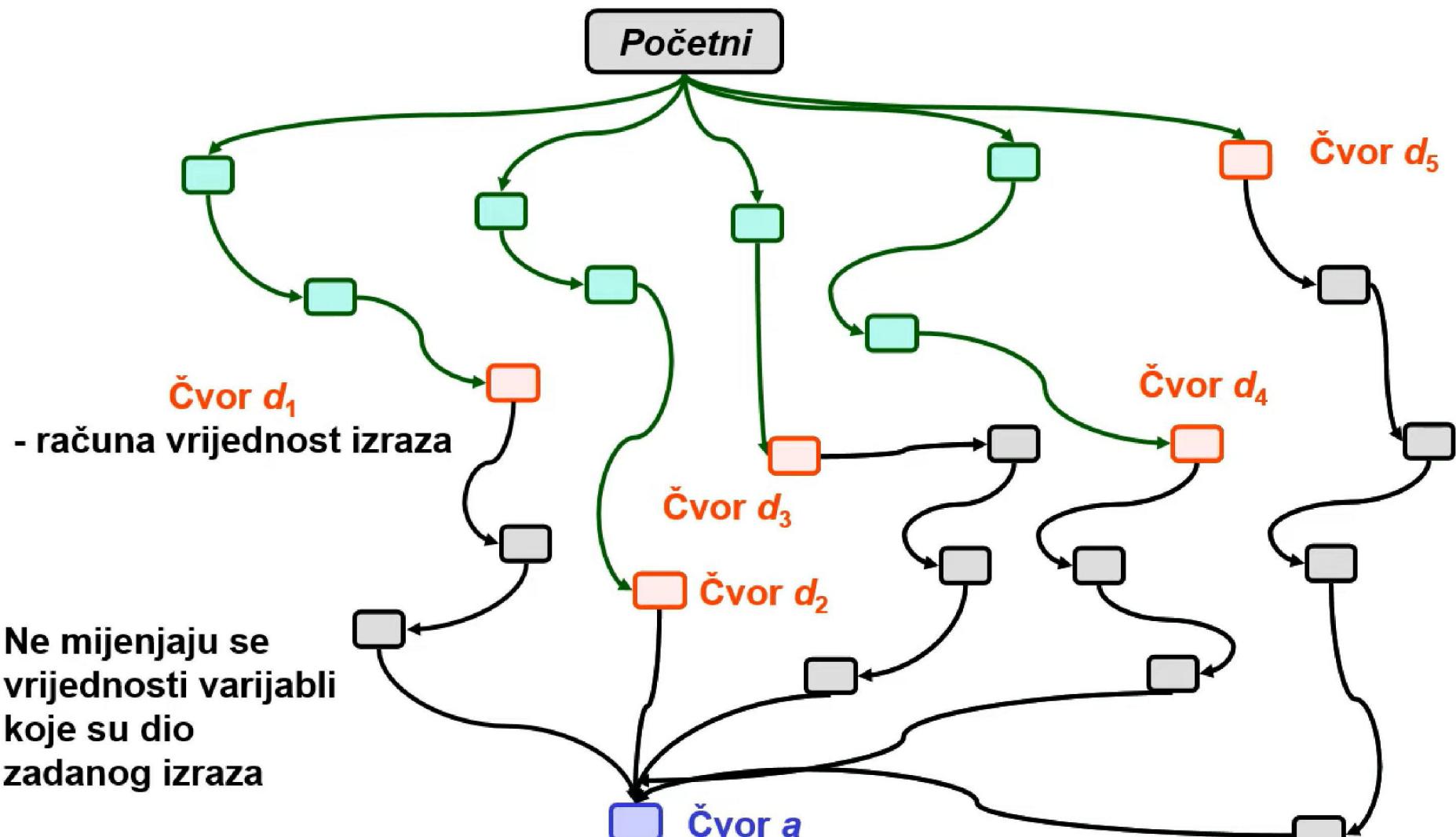
Važeća definicija za uporabu varijable **a** u bloku **B4** su definicije iz blokova **{B1, B2}**



Dostupni izrazi

- **Unaprijedna analiza**
- **Vrijednosti kojih izraza su dostupne - izračunate**
- **Bit vektor**
 - zasebni bitovi dodjeljuju se različitim izrazima

Dostupni izrazi

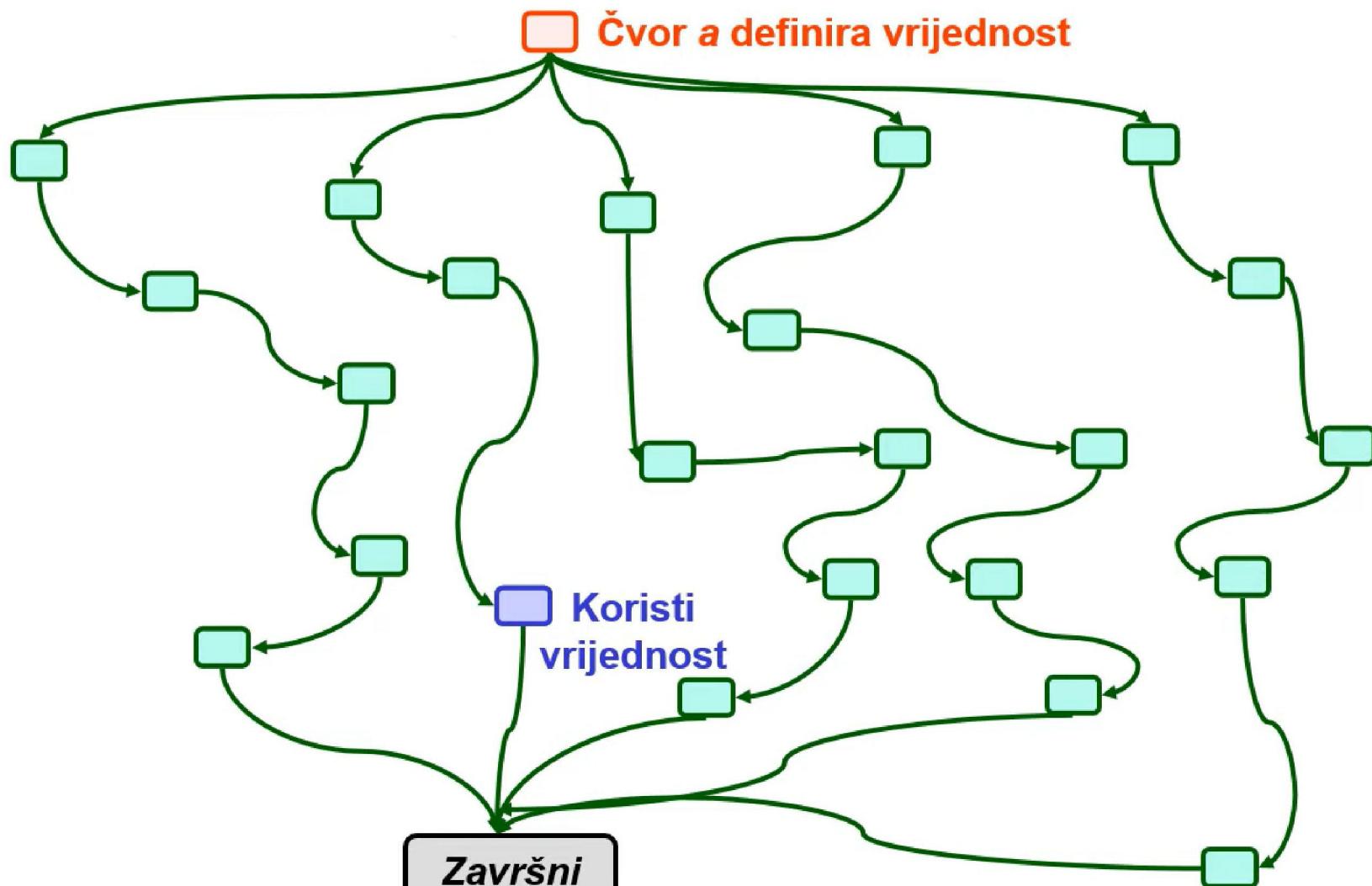


Vrijednost izraza je dostupna (izračunata) u čvoru a

Žive varijable

- Unazadna analiza
- Vrijednosti kojih varijabli se koriste u nastavku programa
- Bit vektor
 - zasebni bitovi dodjeljuju se različitim uporabama varijable

Žive varijable



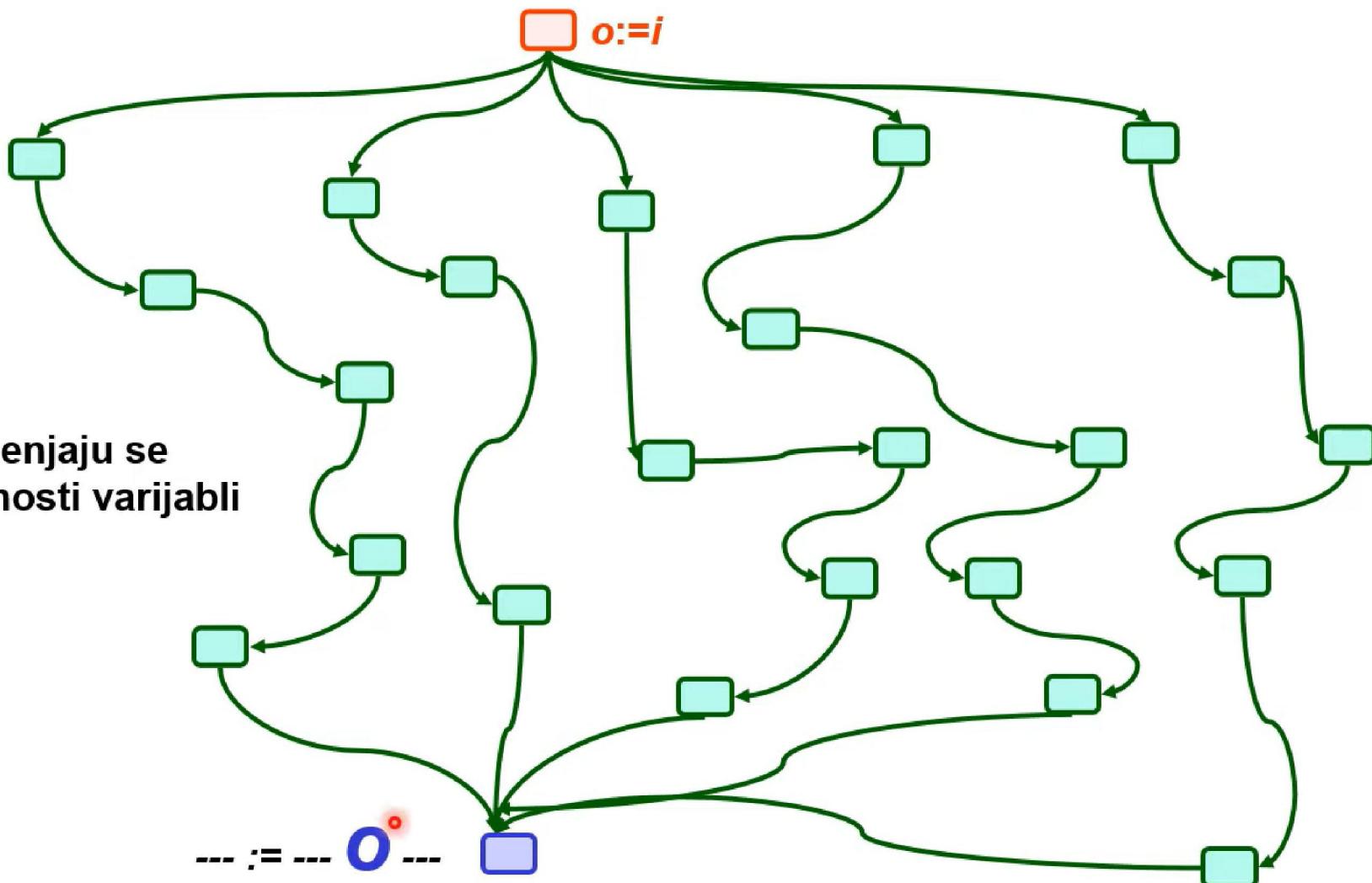
Varijabla je živa u čvoru **a**

Prenošenje naredbe preslikavanja

- Unaprijedna analiza
- Oblik naredbe preslikavanja: $o := i$
- Zamjena uporabe varijable o varijablom i
- Varijabla o je mrtva
- Bit vektor
 - zasebni bitovi dodjeljuju se različitim naredbama preslikavanja

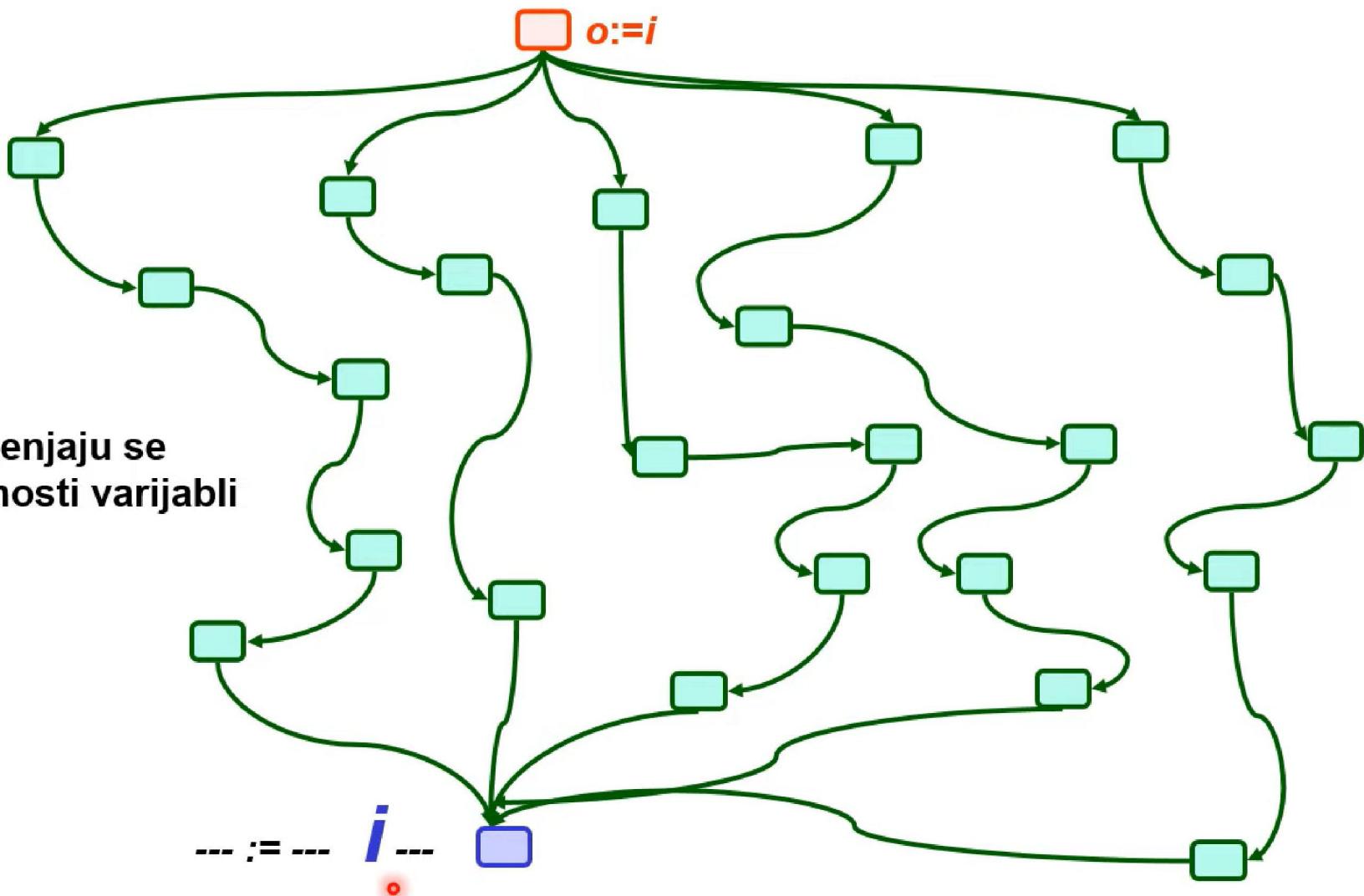
Prenošenje naredbe preslikavanja

Ne mijenjaju se
vrijednosti varijabli
 o i i



Prenošenje naredbe preslikavanja

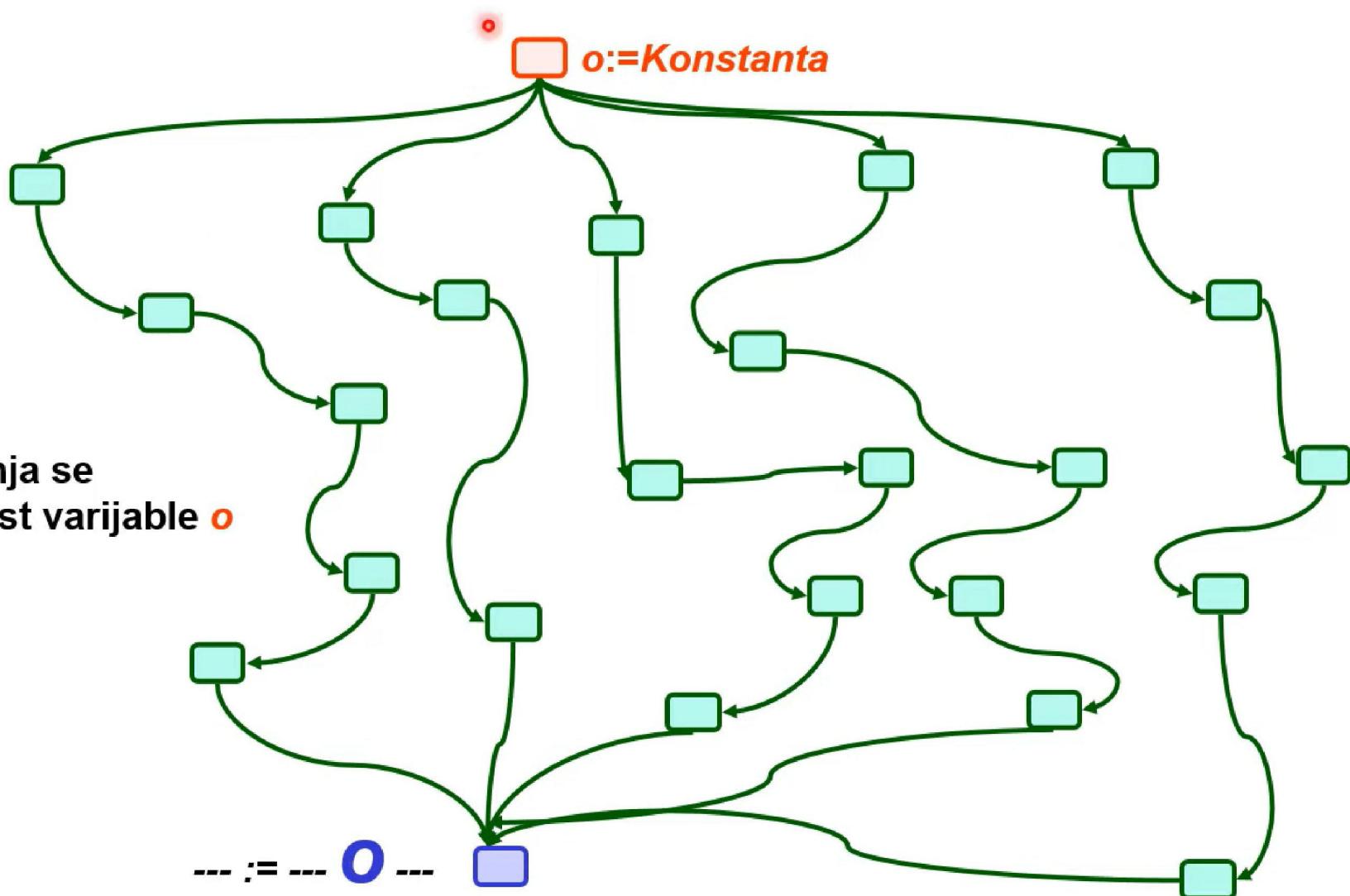
Ne mijenjaju se
vrijednosti varijabli
o i ***i***



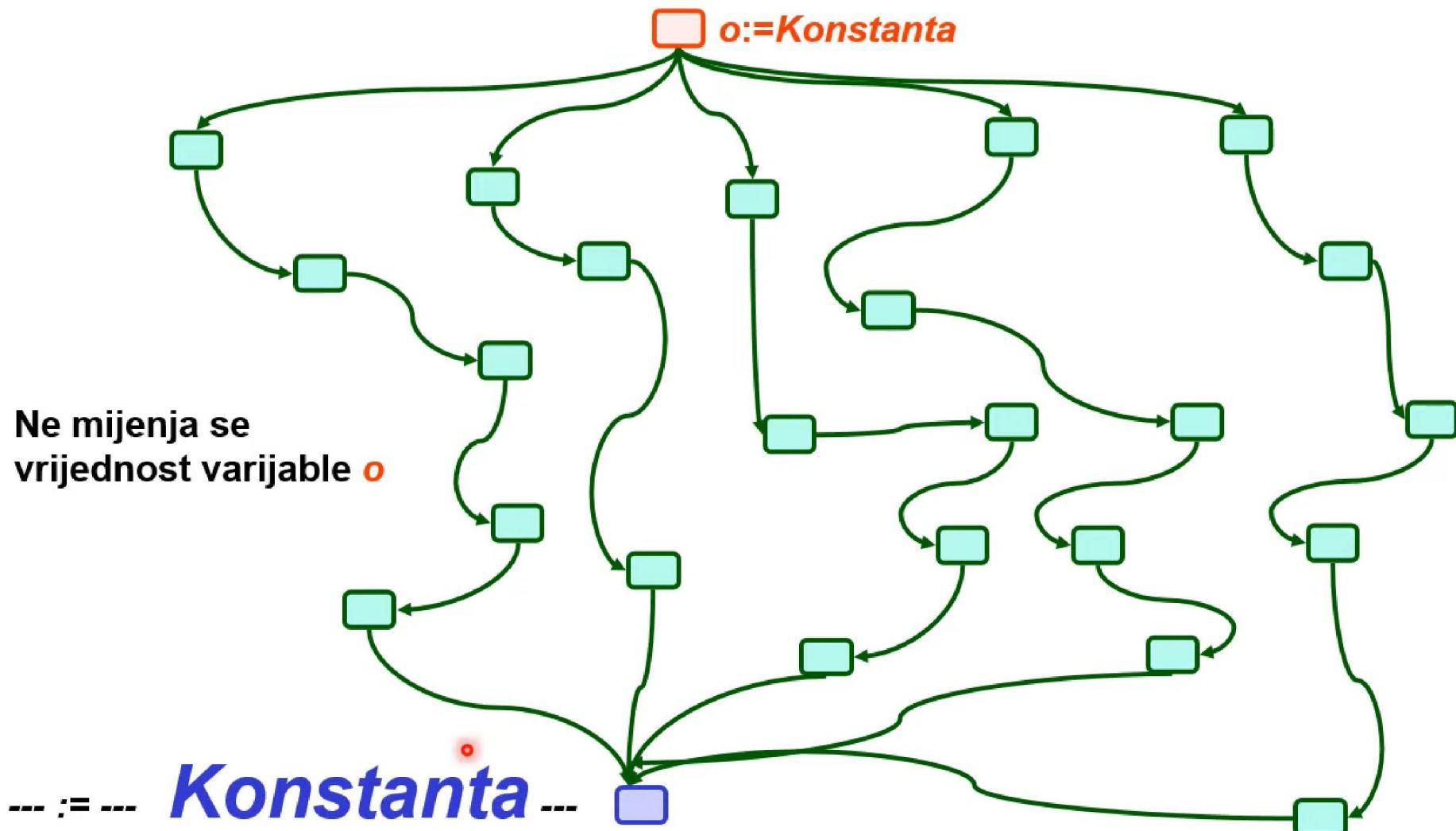
- Unaprijedna analiza
- Oblik naredbe preslikavanja: $o:=Konstanta$
- Zamjena uporabe varijable o s $Konstanta$
- Varijabla o je mrtva
- Konstantni izrazi
- Bit vektor
 - zasebni bitovi dodjeljuju se različitim varijablama

Prenošenje konstante

Ne mijenja se
vrijednost varijable **o**



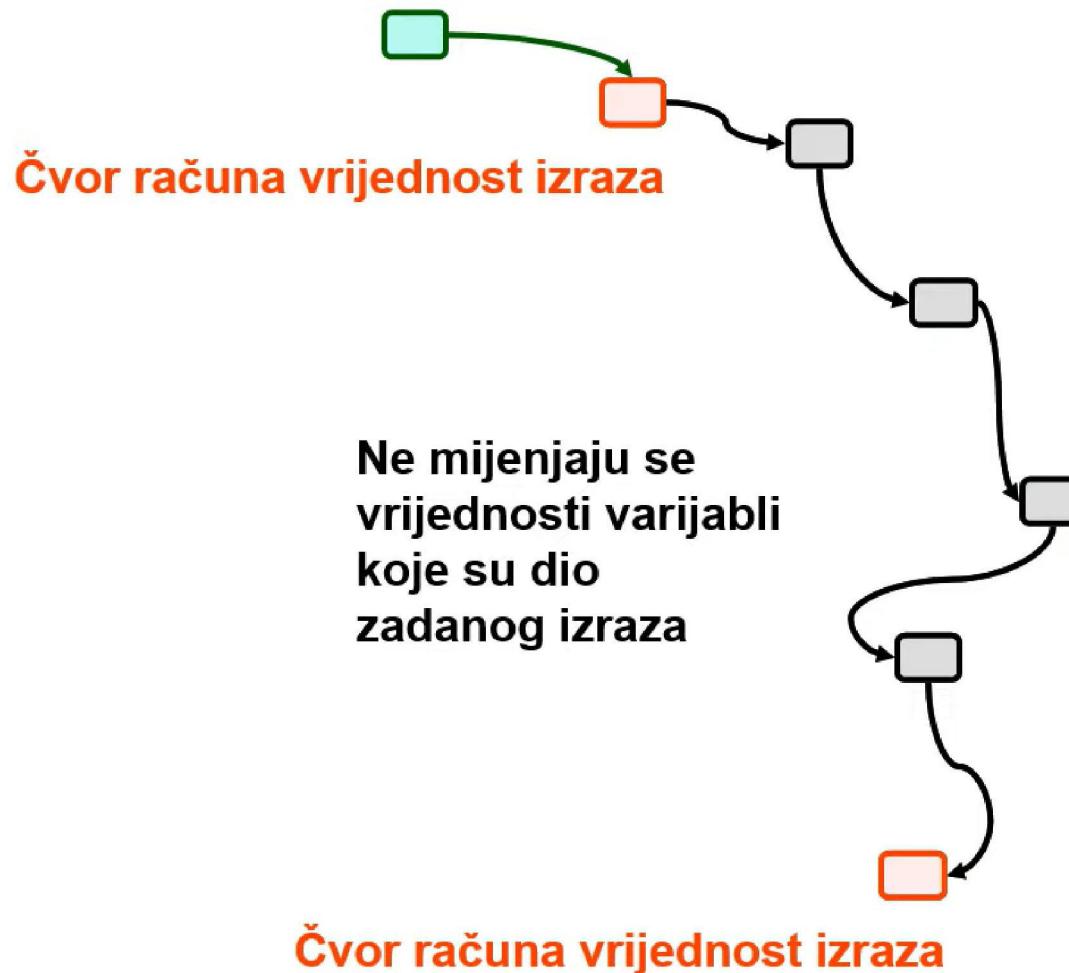
Prenošenje konstante



Višekratno računanje izraza

- **Dvosmjerna analiza**
- **Bit vektor**
 - zasebni bitovi dodjeljuju se različitim izrazima

Višekratno računanje izraza

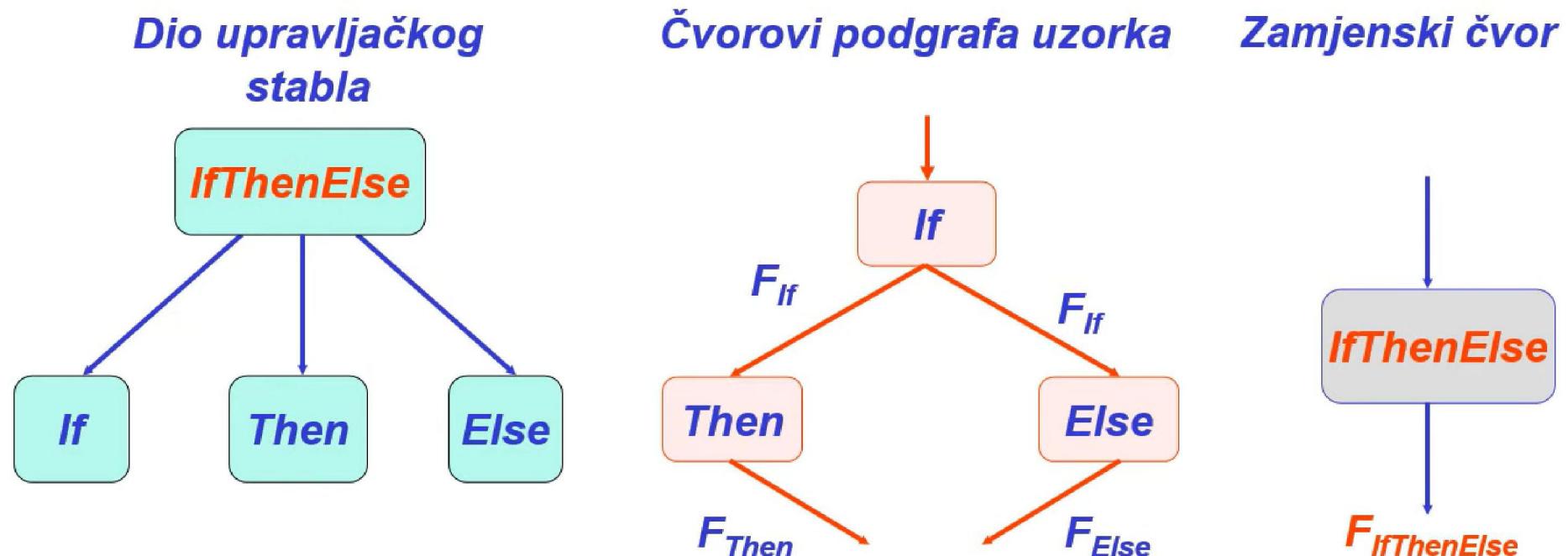


Eliminacijski postupak

- **Koristi se upravljačko stablo**
- **Nedostatak u usporedbi s iterativnim postupkom**
 - složeniji za izvođenje
 - teže ga je programski ostvariti
- **Prednost u usporedbi s iterativnim postupkom**
 - ako postupci optimiranja promijene dio međukôda
 - nije potrebno promijeniti sve jednadžbe toka podataka
- **Jednadžbe toka podataka**
 - obilaze se čvorovi upravljačkog stabla
 - čvorovi se obilaze dva puta
 - prvi prolaz – od listova prema korijenu
 - drugi prolaz – od korijena prema listovima

Eliminacijski postupak

- Prvi prolaz - unaprijedna analiza toka podataka koja računa doseg definicije varijable

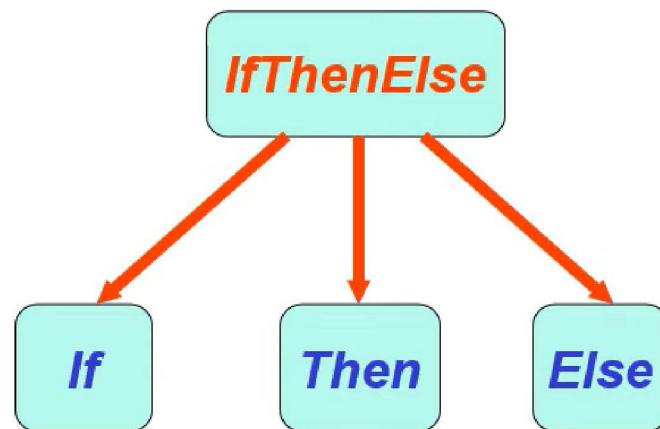


$$F_{\text{IfThenElse}} = (F_{\text{Then}} \circ F_{\text{If}}) \vee (F_{\text{Else}} \circ F_{\text{If}})$$

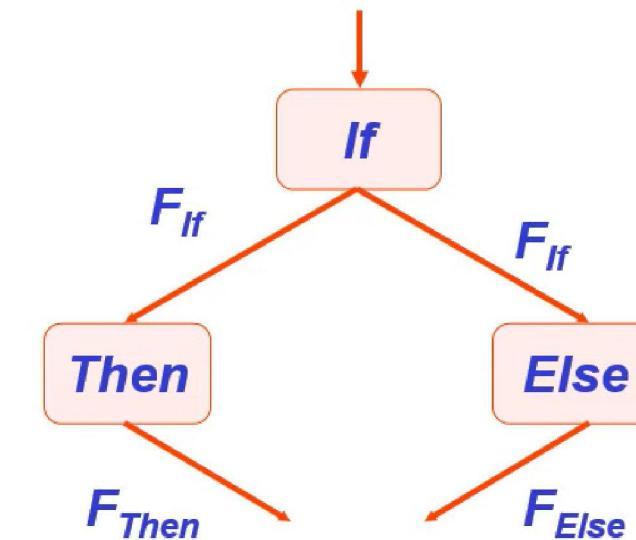
Eliminacijski postupak

- **Drugi prolaz** - unaprijedna analiza toka podataka koja računa doseg definicije varijable

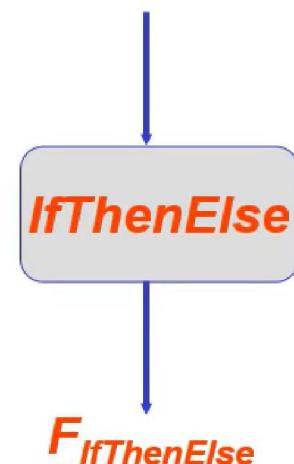
Dio upravljačkog stabla



Čvorovi podgrafa uzorka



Zamjenski čvor



$$\text{Uzaz(If)} = \text{Uzaz(IfThenElse)}$$

$$\text{Uzaz(Then)} = \text{Uzaz(Else)} = F_{if}(\text{Uzaz(If)})$$

Analiza zavisnosti podataka

- **Graf zavisnosti**

- Graf zavisnosti upravljačkog tijeka
- Graf zavisnosti podataka

```
1)      i = j + k;  
2)      ako ( i > 37 )  
3)          skoči L1  
           inac̄e  
           {  
4)              n = j * m;  
5)              m = n + 13;  
           }  
6) L1:   n = m / 23;
```

Analiza zavisnosti podataka

Naredba N1 mijenja vrijednost podatka

Naredba N2 koristi

N1 i N2 \Rightarrow unaprijedno zavisne (f)

Naredba N2 mijenja vrijednost podatka

Naredba N1 koristi

N1 i N2 \Rightarrow unazadno zavisne (a)

Naredba N1 mijenja vrijednost podatka

Naredba N2 mijenja

N1 i N2 \Rightarrow zavisnost odredišta (o)

Naredba N1 koristi vrijednost podatka

Naredba N2 koristi

N1 i N2 \Rightarrow zavisnost izvorišta (i)

$$1) \quad i = j + k;$$

2) ako ($i > 37$)

3) skoči L1

inache

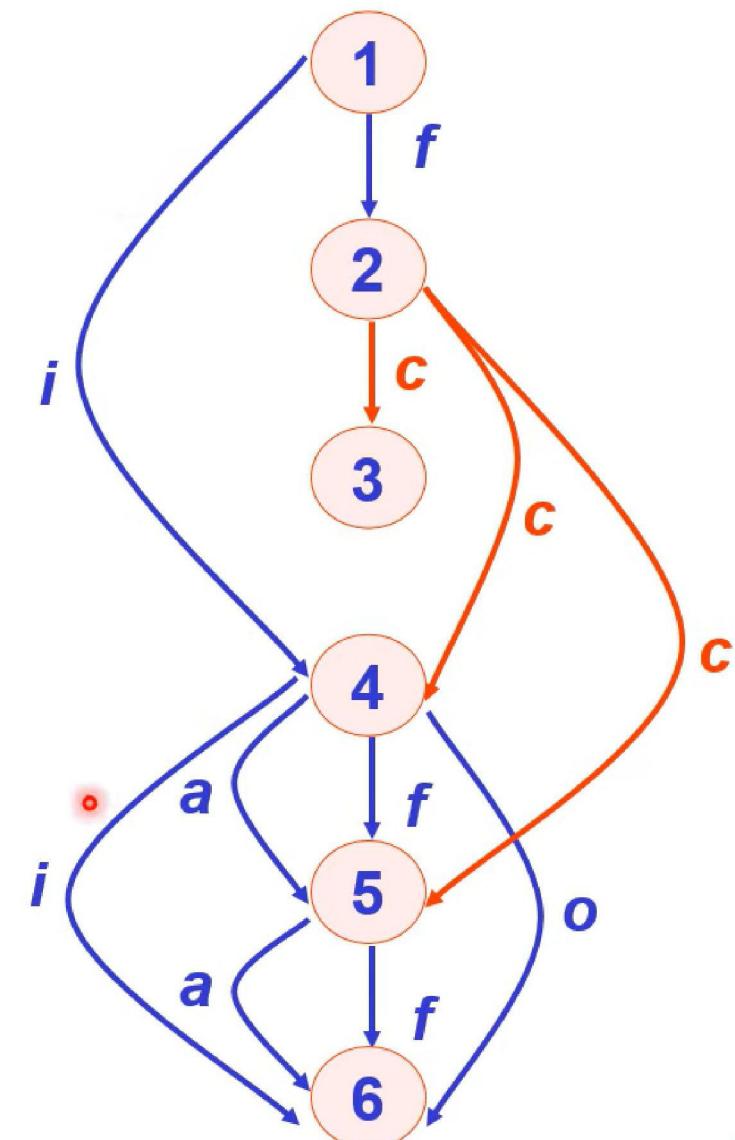
{

$$4) \quad n = j * m;$$

$$5) \quad m = n + 13;$$

}

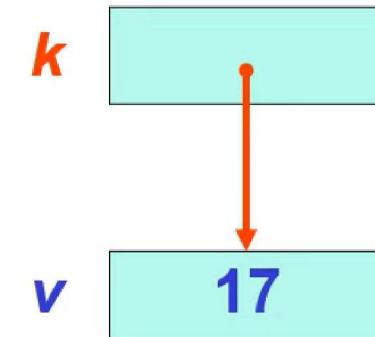
$$6) \quad L1: \quad n = m / 23;$$



- **Pseudonimi**

- različiti načini pristupa istim podatkovnim objektima
- pokazivači u višem programskom jeziku (npr. C, C++)
 - pristup podatku primjenom
 - skalarne varijable
 - pokazivača

```
main()
{
    int *k;
    int v;
    k = &v;
    v = 17;
    printf("%d,%d", *k, v);
}
```



Analiza pseudonima

- **Pseudonimi**

- **C**

- razmjenom vrijednosti pokazivača
 - računanjem indeksa polja
 - rekurzivnim izvođenjem potprograma koji dohvaćaju lokalnu varijablu koja je definirana kao **static**
 - deklaracijom **union**

- **FORTRAN**

- naredba **EQUIVALENCE**
 - definira da dva ili više skupa podataka započinju istom memorijskom adresom

- **Java, Python**

- kopiranjem programskih varijabli kojima se imenuju neprimitivni tipovi podataka

Analiza pseudonima

int *k1, *k2;

int v, z;

P()

{

z = v-19;

***k1 = 31;**

***k2 = 47;**

z = v-19;

}

Analiza pseudonima

Faze analize

*Skupljač
pseudonima*

Faze sinteze

*Prenositelj
pseudonima*

- **Pravila skupljača pseudonima**

- Nailaskom na naredbu ***k=&v***

- pokazivač ***k*** pokazuje isključivo na varijablu ***v***

- relacija ***(k, v)***

- Nailaskom na naredbu ***k1=k2***

- pokazivač ***k1*** pokazuje na sve one varijable na koje pokazuje pokazivač ***k2***

- relacija ***(k2, x) → (k1, x)***

Analiza pseudonima

- **Prenositelj pseudonima**

- unaprijedna analiza toka podataka
- obilazi graf tijeka izvođenja programa od početnog do završnog čvora

Jednadžbe toka podataka

$Izlaz(k=&v) =$

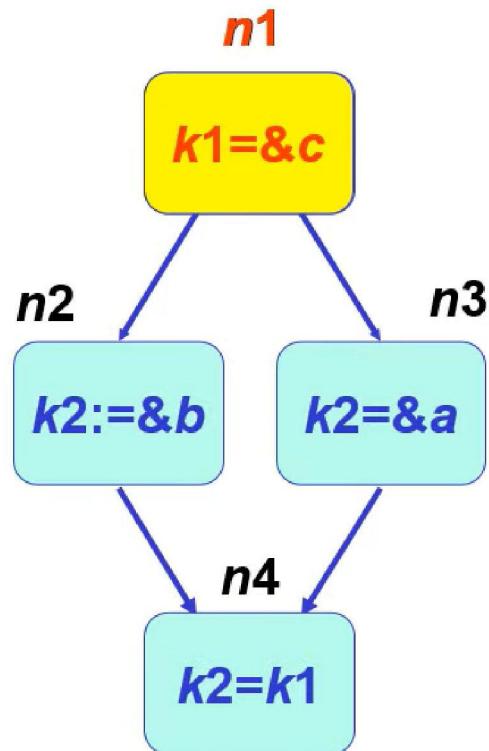
$$(Ulaz(k=&v) \setminus \{(k,x) \mid \forall x\}) \cup \{(k,v)\}$$

$Izlaz(k_1=k_2) =$

$$(Ulaz(k_1=k_2) \setminus \{(k_1,x) \mid \forall x\}) \cup \{(k_1,y) \mid (k_2,y) \in Ulaz(k_1=k_2)\}$$

$Ulaz(n) = \bigcup_{p \in P} Izlaz(p),$ gdje je P skup svih naredbi koje
• neposredno prethode naredbi n

Analiza pseudonima



**Skupljač
pseudonima**

$(k1, c)$

**Prenositelj
pseudonima**

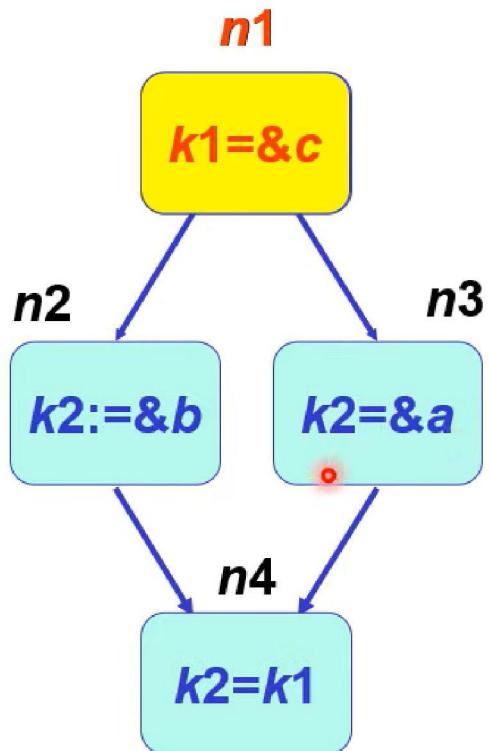
$Ulaz(n1) = \{ \}$

$Izlaz(n1) =$

$(Ulaz(n1) \setminus \{(k1, x) | \forall x\}) \cup \{(k1, c)\} =$
 $(\{ \} \setminus \{ \}) \cup \{(k1, c)\} =$
 $\{(k1, c)\}$

•

Analiza pseudonima



*Skupljač
pseudonima*

$(k1, c)$

*Prenositelj
pseudonima*

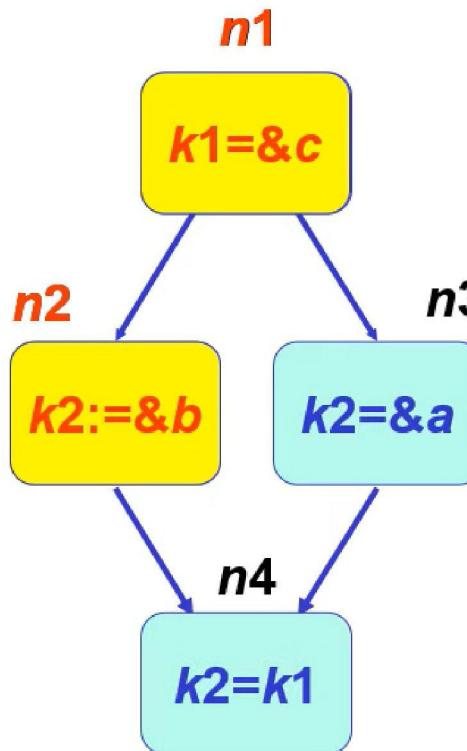
$Ulaz(n1) = \{ \}$

$Izlaz(n1) = \{(k1, c)\}$

$Ulaz(n2) = Izlaz(n1) = \{(k1, c)\}$

$Ulaz(n3) = Izlaz(n1) = \{(k1, c)\}$

Analiza pseudonima



**Skupljač
pseudonima**

($k1, c$)

($k2, b$)

**Prenositelj
pseudonima**

$Ulaz(n1) = \{ \}$

$Izlaz(n1) = \{(k1, c)\}$

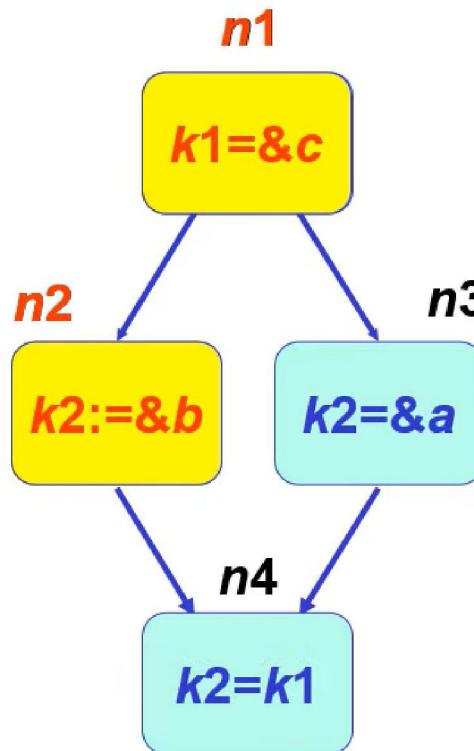
$Ulaz(n2) = Izlaz(n1) = \{(k1, c)\}$

$Ulaz(n3) = Izlaz(n1) = \{(k1, c)\}$

$Izlaz(n2) =$

$(Ulaz(n2) \setminus \{(k2, x) | \forall x\}) \cup \{(k2, b)\} =$
 $(\{(k1, c)\} \setminus \{ \}) \cup \{(k2, b)\} =$
 $\{(k1, c), (k2, b)\}$

Analiza pseudonima



*Skupljač
pseudonima*

(k_1, c)

(k_2, b)

*Prenositelj
pseudonima*

$Ulaz(n_1) = \{ \}$

$Izlaz(n_1) = \{(k_1, c)\}$

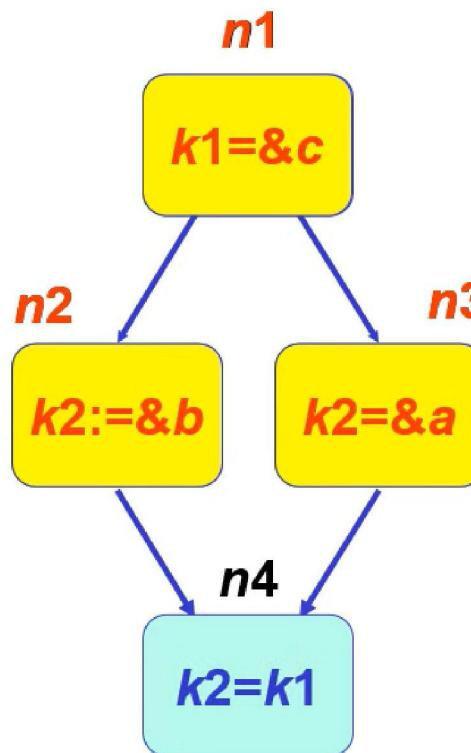
$Ulaz(n_2) = Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_3) = Izlaz(n_1) = \{(k_1, c)\}$

$Izlaz(n_2) = \{(k_1, c), (k_2, b)\}$

•

Analiza pseudonima



Skupljač pseudonima

($k1, c$)

($k2, b$)
($k2, a$)

Prenositelj pseudonima

$Ulaz(n1) = \{ \}$

$Izlaz(n1) = \{(k1, c)\}$

$Ulaz(n2) = Izlaz(n1) = \{(k1, c)\}$

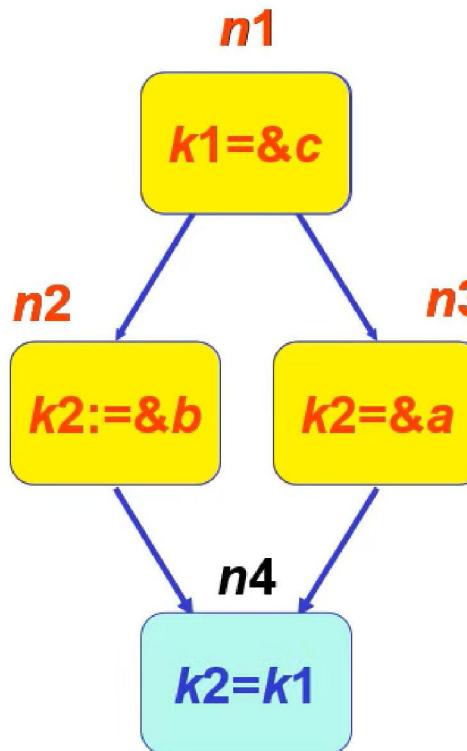
$Ulaz(n3) = Izlaz(n1) = \{(k1, c)\}$

$Izlaz(n2) = \{(k1, c), (k2, b)\}$

$Izlaz(n3) = \bullet$

$(Ulaz(n3) \setminus \{(k2, x) | \forall x\}) \cup \{(k2, a)\} =$
 $\{ \} \setminus \{ \} \cup \{(k2, a)\} =$
 $\{(k1, c), (k2, a)\}$

Analiza pseudonima



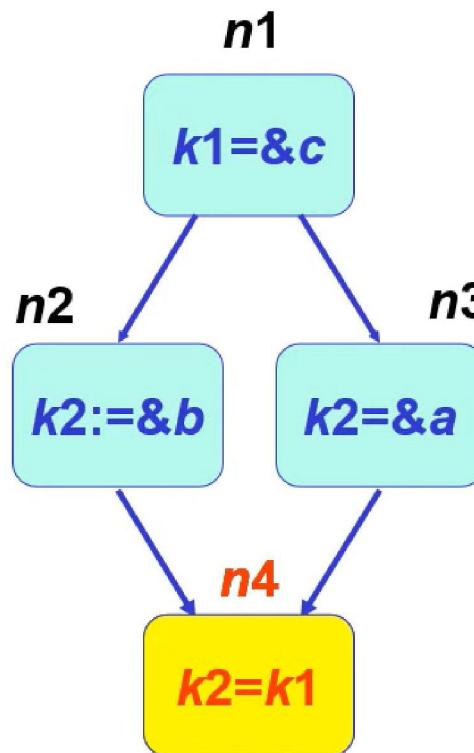
*Skupljač
pseudonima*

$(k1,c)$
 $(k2,b)$
 $(k2,a)$

*Prenositelj
pseudonima*

$Ulaz(n1) = \{ \}$
 $Izlaz(n1) = \{(k1,c)\}$
 $Ulaz(n2) = Izlaz(n1) = \{(k1,c)\}$
 $Ulaz(n3) = Izlaz(n1) = \{(k1,c)\}$
 $Izlaz(n2) = \{(k1,c), (k2,b)\}$
 $Izlaz(n3) = \{(k1,c), (k2,a)\}$

Analiza pseudonima



**Skupljač
pseudonima**

(k_1, c)

(k_2, b)
(k_2, a)

**Prenositelj
pseudonima**

$Ulaz(n_1) = \{ \}$

$Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_2) = Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_3) = Izlaz(n_1) = \{(k_1, c)\}$

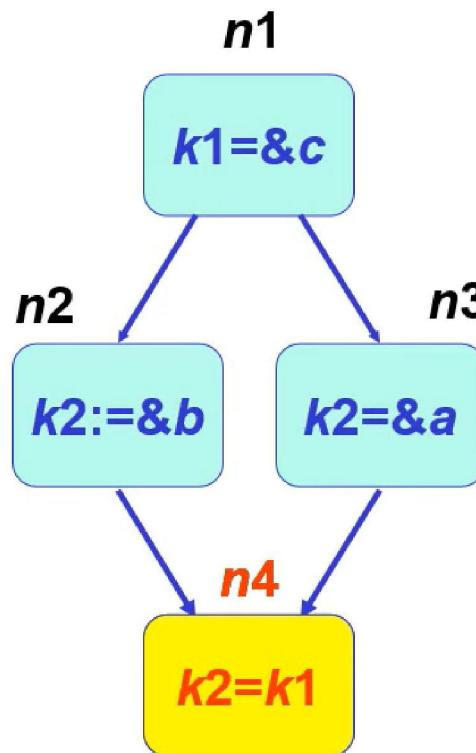
$Izlaz(n_2) = \{(k_1, c), (k_2, b)\}$

$Izlaz(n_3) = \{(k_1, c), (k_2, a)\}$

$Ulaz(n_4) =$

$Izlaz(n_2) \cup Izlaz(n_3) =$
 $\{(k_1, c), (k_2, b)\} \cup \{(k_1, c), (k_2, a)\} =$
 $\{(k_1, c), (k_2, b), (k_2, a)\}$

Analiza pseudonima



**Skupljač
pseudonima**

($k1, c$)

($k2, b$)
($k2, a$)

**Prenositelj
pseudonima**

$Ulaz(n1) = \{ \}$

$Izlaz(n1) = \{(k1, c)\}$

$Ulaz(n2) = Izlaz(n1) = \{(k1, c)\}$

$Ulaz(n3) = Izlaz(n1) = \{(k1, c)\}$

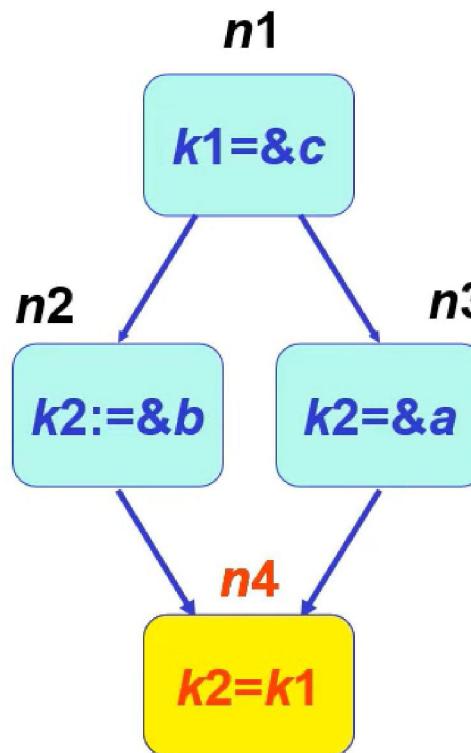
$Izlaz(n2) = \{(k1, c), (k2, b)\}$

$Izlaz(n3) = \{(k1, c), (k2, a)\}$

$Ulaz(n4) = \{(k1, c), (k2, b), (k2, a)\}$

•

Analiza pseudonima



**Skupljač
pseudonima**

(k_1, c)

(k_2, b)
(k_2, a)

(k_1, x) → (k_2, x)

**Prenositelj
pseudonima**

$Ulaz(n_1) = \{ \}$

$Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_2) = Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_3) = Izlaz(n_1) = \{(k_1, c)\}$

$Izlaz(n_2) = \{(k_1, c), (k_2, b)\}$

$Izlaz(n_3) = \{(k_1, c), (k_2, a)\}$

$Ulaz(n_4) = \{(k_1, c), (k_2, b), (k_2, a)\}$

$Izlaz(n_4) =$

$(Ulaz(n_4) \setminus \{(k_2, x) | \forall x\}) \cup$

$\{(k_2, y) | (k_1, y) \in Ulaz(n_4)\} =$

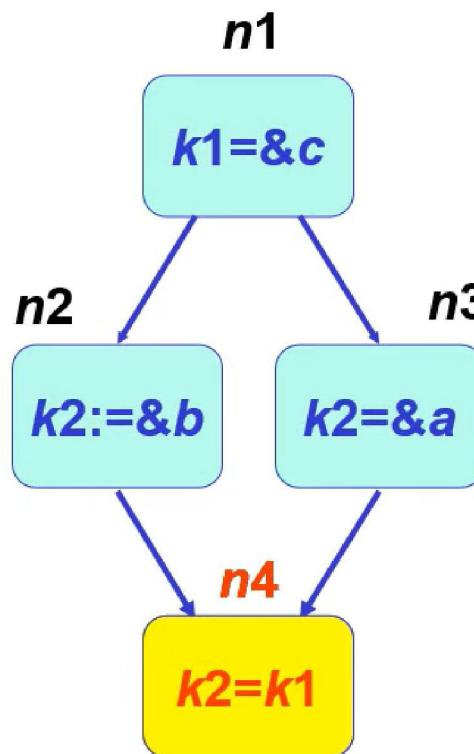
$\{(k_1, c), (k_2, b), (k_2, a)\} \setminus$

$\{(k_2, b), (k_2, a)\} \cup \{(k_2, c)\} =$

$\{(k_1, c)\} \cup \{(k_2, c)\} =$

$\{(k_1, c), (k_2, c)\}$

Analiza pseudonima



**Skupljač
pseudonima**

(k_1, c)

(k_2, b)
 (k_2, a)

$(k_1, x) \rightarrow (k_2, x)$

**Prenositelj
pseudonima**

$Ulaz(n_1) = \{ \}$

$Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_2) = Izlaz(n_1) = \{(k_1, c)\}$

$Ulaz(n_3) = Izlaz(n_1) = \{(k_1, c)\}$

$Izlaz(n_2) = \{(k_1, c), (k_2, b)\}$

$Izlaz(n_3) = \{(k_1, c), (k_2, a)\}$

$Ulaz(n_4) = \{(k_1, c), (k_2, b), (k_2, a)\}$

$Izlaz(n_4) = \{(k_1, c), (k_2, c)\}$

•