

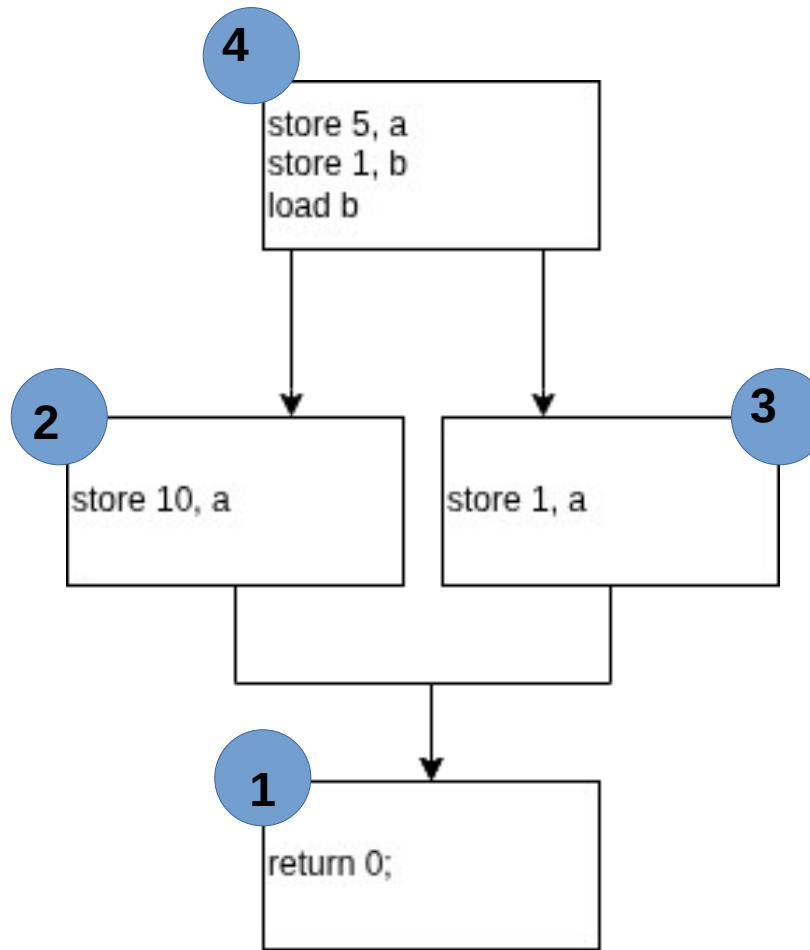
# **Dead store elimination**

Angelina Jordanov

- DSE – uklanja naredbe upisa u memoriju nakon kojih se ta vrednost nikada ne ucitava
- Cilj: smanjiti vreme izvršavanja
- Zasniva se na **analizi zivih promenljivih**
- Promenljiva je ziva ako ce njena vrednost biti koriscena u buducnosti
- Smer: **unazad**

# Control Flow Graph

```
1 #include <stdio.h>
2
3 int main(){
4
5     int a = 5;
6     int b = 1;
7
8     if(b > 0)
9         a = 10;
10    else
11        a = 1;
12
13
14    return 0;
15 }
```



loads\_in[BB] – zive na pocetku bloka

loads\_out[BB] – zive na kraju bloka

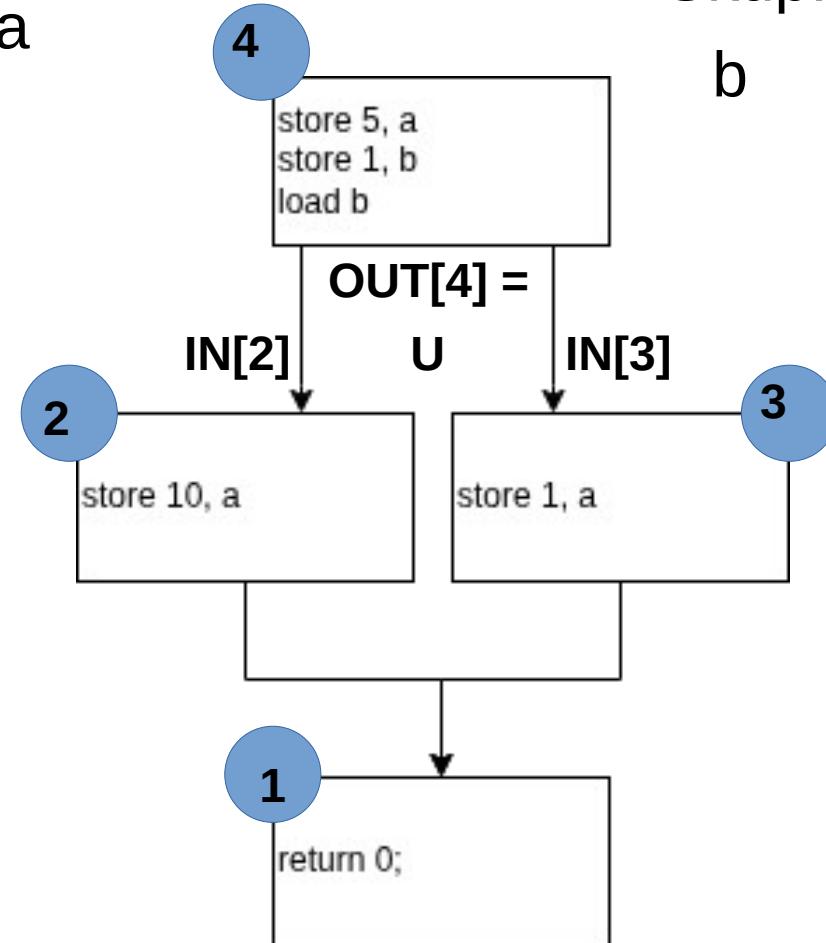
**loads\_out[BB] =  $\bigcup$  loads\_in[succ]**

**Init:** loads\_in[BB] = loads\_out[BB]

Dokle god ne dodjemo do kraja instrukcija radi:

- ako se naidje na **load**, staviti **promenljivu u skup**
- ako se naidje na **store** i ako je ta promenljiva u skupu **izbaciti je iz skupa**

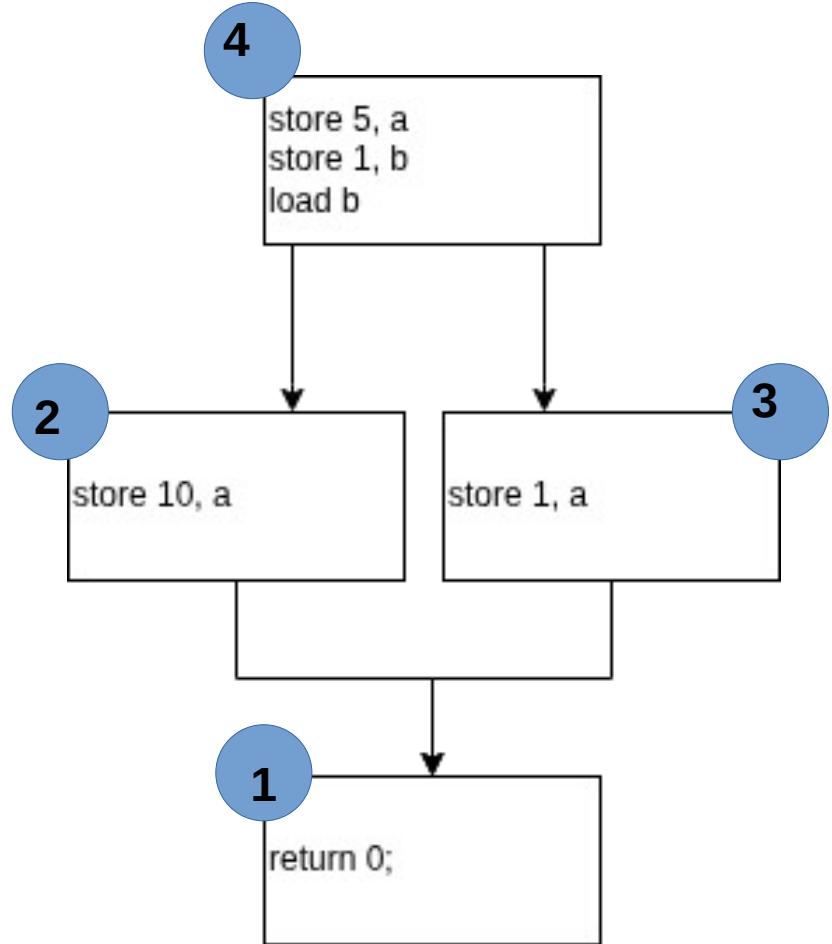
Skup:  
b



## Analiza bloka:

- new\_out = U loads\_in[succ]
- Ako se razlikuje od loads\_out:
  - apdejtuj loads\_out
  - setuj fleg da nije stabilno stanje
- Izracunaj new\_in
- Ako se razlikuje od loads\_in:
  - apdejtuj loads\_in
  - setuj fleg da nije stabilno stanje

Ponavljati dokle god nije stabilno stanje.



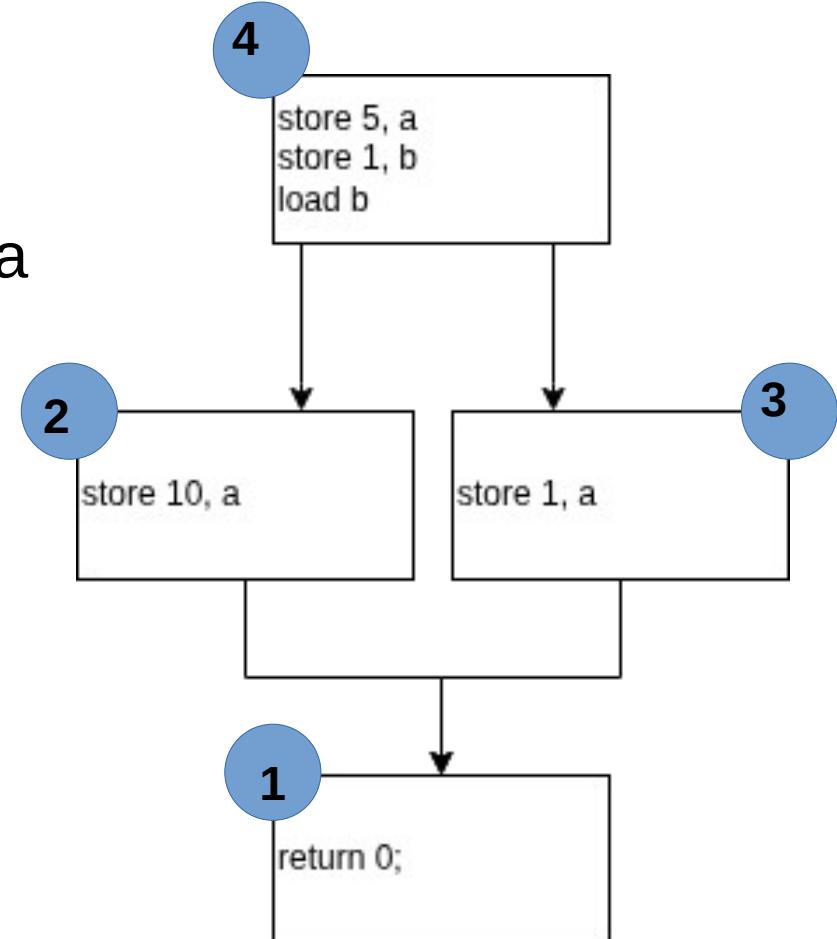
## Faza brisanja:

- redosled obilaska blokova nije bitan

**Init:** loads\_in[BB] = loads\_out[BB]

Dokle god ne dodjemo do kraja instrukcija radi:

- ako se naidje na **load**, staviti **promenljivu u skup**
- ako se naidje na **store**:
  - i ako je u skupu **izbaciti je iz skupa**
    - ako nije u skupu izbaciti instrukciju **dead store detected**



```
; Function Attrs: noinline nounwind optnone uwtable
define dso_local i32 @main() #0 {
    %1 = alloca i32, align 4
    %2 = alloca i32, align 4
    %3 = alloca i32, align 4
    store i32 0, ptr %1, align 4
    store i32 5, ptr %2, align 4
    store i32 1, ptr %3, align 4
    %4 = load i32, ptr %3, align 4
    %5 = icmp sgt i32 %4, 0
    br i1 %5, label %6, label %7

6:                                         ; preds = %0
    store i32 10, ptr %2, align 4
    br label %8

7:                                         ; preds = %0
    store i32 1, ptr %2, align 4
    br label %8

8:                                         ; preds = %7, %6
    ret i32 0
}
```

**BEFORE**

```
; Function Attrs: noinline nounwind optnone uwtable
define dso_local i32 @main() #0 {
    %1 = alloca i32, align 4
    %2 = alloca i32, align 4
    %3 = alloca i32, align 4
    store i32 1, ptr %3, align 4
    %4 = load i32, ptr %3, align 4
    %5 = icmp sgt i32 %4, 0
    br i1 %5, label %6, label %7

6:                                         ; preds = %0
    br label %8

7:                                         ; preds = %0
    br label %8

8:                                         ; preds = %7, %6
    ret i32 0
}
```

**AFTER**

## Ogranicenja:

- ne radi u slucaju alias analize
- ne razmatra funkcijске pozive
- ne razmatra instrukcije koje implicitno menjaju memoriju (npr. memset, memcpy)

```
int x;  
int *a = &x;  
int *b = &x;  
  
*a = 5;    \\ store  
printf(*b); \\ load
```