Računanje du lanaca u tehnici testiranja zasnovanoj na toku podataka

Predstavljanje programa međukodom

- Međukod je interna forma programa za potrebe analiziranja i generisanja mašinskog koda
- Može uzeti različite forme
- Mi ćemo razmatrati formu u vidu apstraktnog asemblerskog koda
  - Složeni izrazi se dekomponuju na troadresne naredbe (dst := src1 OP src2)
  - Naredbe za kontrolu toka (while, case) na uslovni goto
  - Poziv procedure: priprema parametara (na stek), call, na kraju return
  - itd

Primer

- Programskom fragmentu:

```
prod = 0;
i = 1;
do {
      prod = prod + a[i] * b[i];
      i = i + 1;
} while ( i <= 20 );
```

Primer

- Odgovara sledeći međukod:
  1. prod = 0
  2. i = 1
  3. t1 = 4 * i
  4. t2 = a[t1]
  5. t3 = 4 * i
  6. t4 = b[t3]
  7. t5 = t2 * t4
  8. t6 = prod + t5
  9. prod = t6
  10. t7 = i + 1
  11. i = t7
  12. if i <= 20 goto 3

Dekompozicija programa na osnovne blokove

- **Osnovni blok** je skup iskaza međukoda koji se izvršavaju u sekvenci (bez grananja) izuzev krajnjeg iskaza u bloku koji može biti uslovni ili bezuslovni skok.

Dekompozicija programa na osnovne blokove

- Dekompozicija programa na osnovne blokove:
  - Odrediti vođe (prve iskaze) osnovnih blokova: prvi iskaz u programu, iskazi koji predstavalju odredišta uslovnih ili bezuslovnih skokova i iskazi koji neposredno slede bezuslovne ili uslovne skokove.
  - Svakom vođi pridružiti iskaze koji ga slede do prvog sledećeg vođe (isključujući taj iskaz) ili do kraja programa.

Primer dekompozicije na osnovne blokove

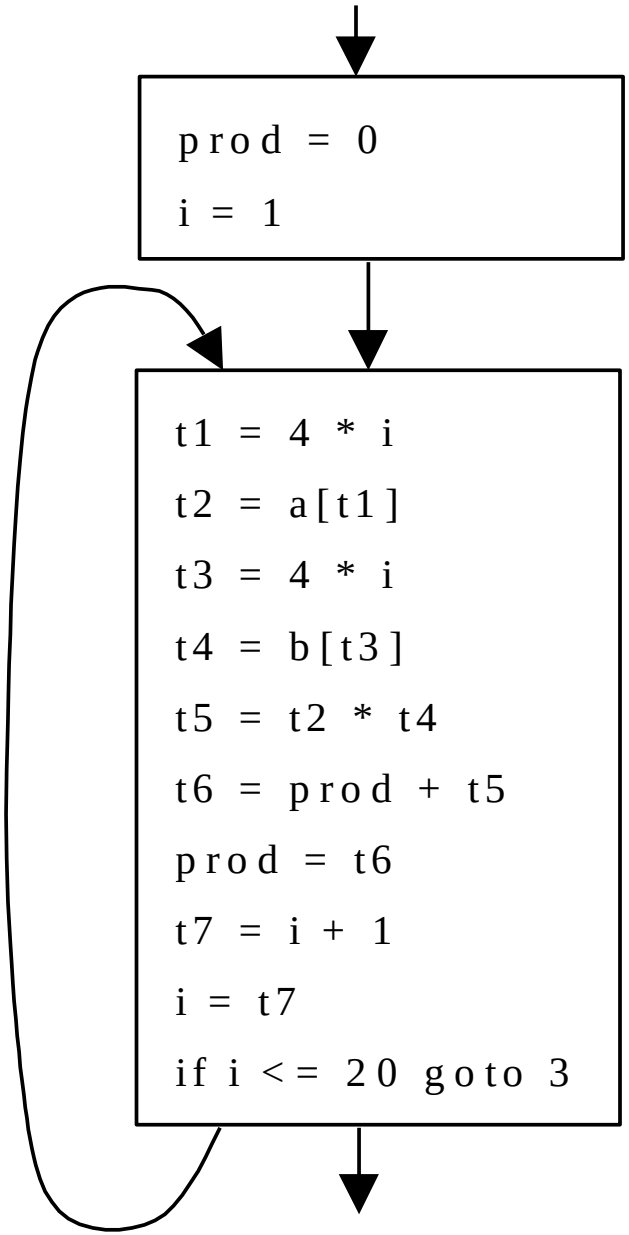
```
1. prod = 0
2. i = 1
```

```
3. t1 = 4 * i
4. t2 = a[t1]
5. t3 = 4 * i
6. t4 = b[t3]
7. t5 = t2 * t4
8. t6 = prod + t5
9. prod = t6
10. t7 = i + 1
11. i = t7
12.if i <= 20 goto 3
```

Graf toka kontrole na nivou osnovnih blokova

- U **grafu toka kontrole**, osnovni blokovi predstavljaju čvorove grafa.
- Od bloka B1 postoji orijetisana grana ka bloku B2, ako B2 neposredno sledi B1 u nekoj izvršnoj sekvenci, odnosno ako:
  - postoji uslovan ili bezuslovan skok od poslednjeg iskaza B1 ka vođi B2, ili
  - B2 neposredno sledi B1 u međukodu, a B1 nema na kraju bezuslovni skok.

Primer

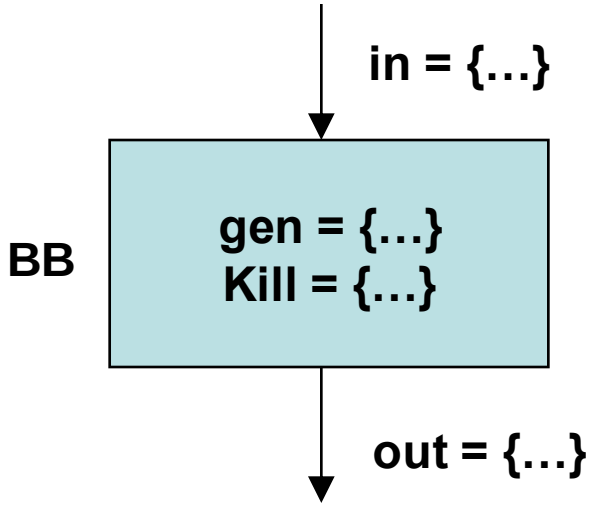

prod = 0
i = 1
t1 = 4 * i
t2 = a[t1]
t3 = 4 * i
t4 = b[t3]
t5 = t2 * t4
t6 = prod + t5
prod = t6
t7 = i + 1
i = t7
if i <= 20 goto 3

Globalna iterativna analiza toka podataka

- Globalna:
  - Sprovodi se nad grafom toka kontrole
  - Cilj = sakupiti informacije na ulasku i izlasku iz osnovnih blokova
- Iterativna:
  - Konstruišu se jednačine koje opisuju tok podataka kroz svaki osnovni blok
    - Rešavaju se u iteracijama, postepenim konvergiranjem ka krajnjem rešenju

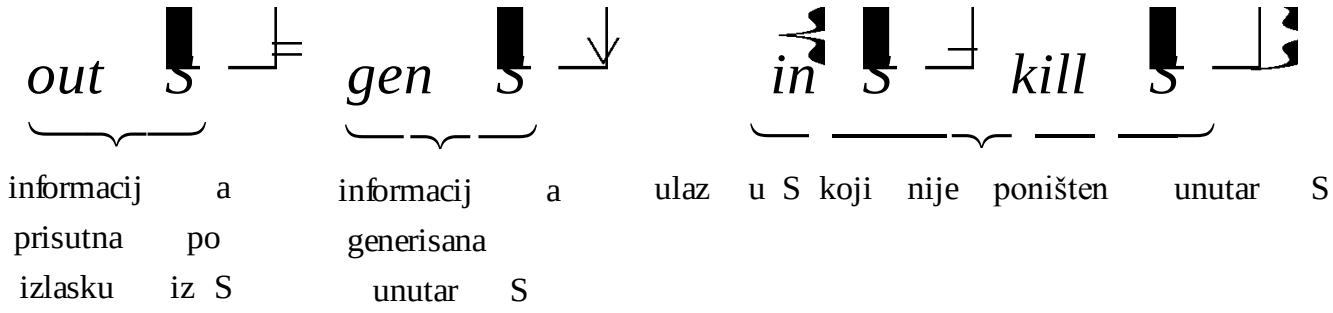
Globalna iterativna analiza toka podataka

- Komponente jednačina toka podataka
  - Skupovi koji sadrže informacije koje se skupljaju
    - Skup **in**: sadrži informacije koje ulaze u blok iz prethodnika u grafu
    - Skup **gen**: informacije koje se stvaraju tj. sakupljaju unutar BB
    - Skup **kill**: informacije koje, na osnovu naredbi unutar BB, utiču na tok informacija kroz BB sa ulaza na izlaz (blokiraju neke)
    - Skup **out** set: informacije na izlazu iz BB



Globalna iterativna analiza toka podataka

- Jednačine nad opisanim skupovima
  - **Jednačina prenosa** opisuje kako se informacija menja dok prolazi kroz osnovni blok:



  - **Jednačine susretanja** opisuju kombinovanje informacija pri spajanju različitih putanja:

$$in[S] = \bigcup_{P \text{ je prethodnik od } S} out\,[P]$$

Globalna iterativna analiza toka podataka

- Algoritam
  - Koristi se iterativni algoritam nepokretne tačke.
  - Zavisno od vrste problema koji se rešava, svaki osnovni blok se obrađuje unapred (od ulaza ka izlazu) ili unazad.
    - Redosled posećivanja osnovnih blokova u grafu nije važan sa stanovišta korektnosti algoritma, jedino iz razloga efikasnosti.

```
Initialize gen and kill sets
Initialize in or out sets (depending on "direction")
while there are no changes in in and out sets {
    for each BB {
      evaluate meet equation to obtain new in[BB]
      evaluate transfer equation to obtain new out[BB]
    }
}
```

Podsećanje - život promenljive

- Za promenljivu X kaže se da je **živa** u iskazu S1, akko
  - X je definisana u nekom iskazu S i
  - postoji putanja od S do S1 koja ne sadrži novu dodelu promenljivoj X.

Lanac Dodele-upotrebe (DU lanac)

- Označava se sa [X,S,S1], gde su
  - S i S1 iskazi, a X promenljiva tako da važi
  - X-u se dodeljuje vrednost u iskazu S i
  - Upotrebljava se vrednost X u iskazu S1
  - X ima dodelu u iskazu S koja je živa u iskazu S1.

Računanje DU lanaca globalnom iterativnom analizom toka podataka

- Za osnovni blok S:
- **in** skup: inicijalno prazan; na kraju sadrži definicije promenljivih koje nisu redefinisane na ulasku u S
- **out** skup: inicijalno prazan; na kraju sadrži definicije promenljivih koje nisu redefinisane na izlasku iz S

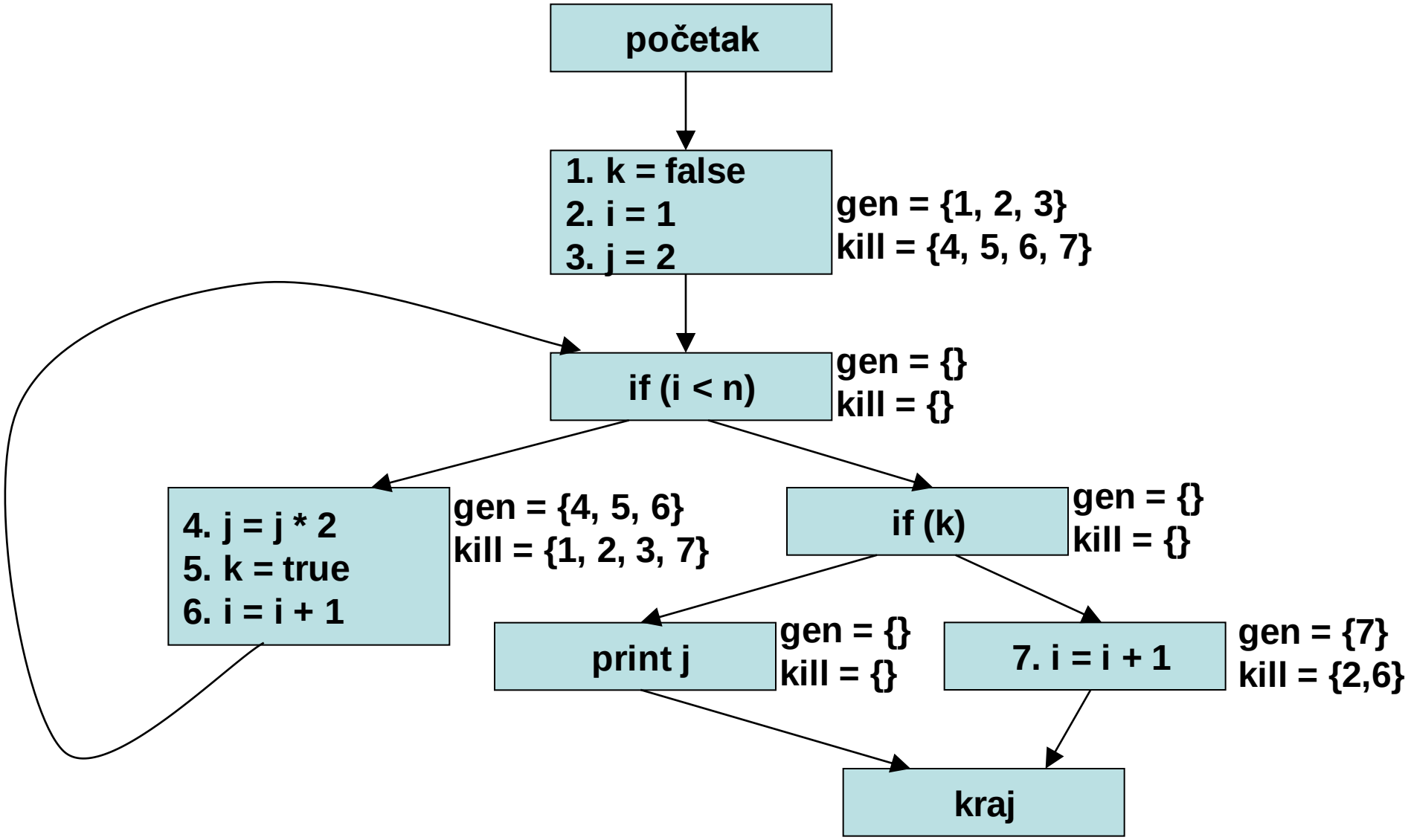
Računanje DU lanaca globalnom iterativnom analizom toka podataka

- Za osnovni blok S:
- **gen** = { bi | bi je definicija u S }
- **kill** = { bi | ako je bi dodela promenljivoj x a u S postoji druga dodela vrednost x-u}

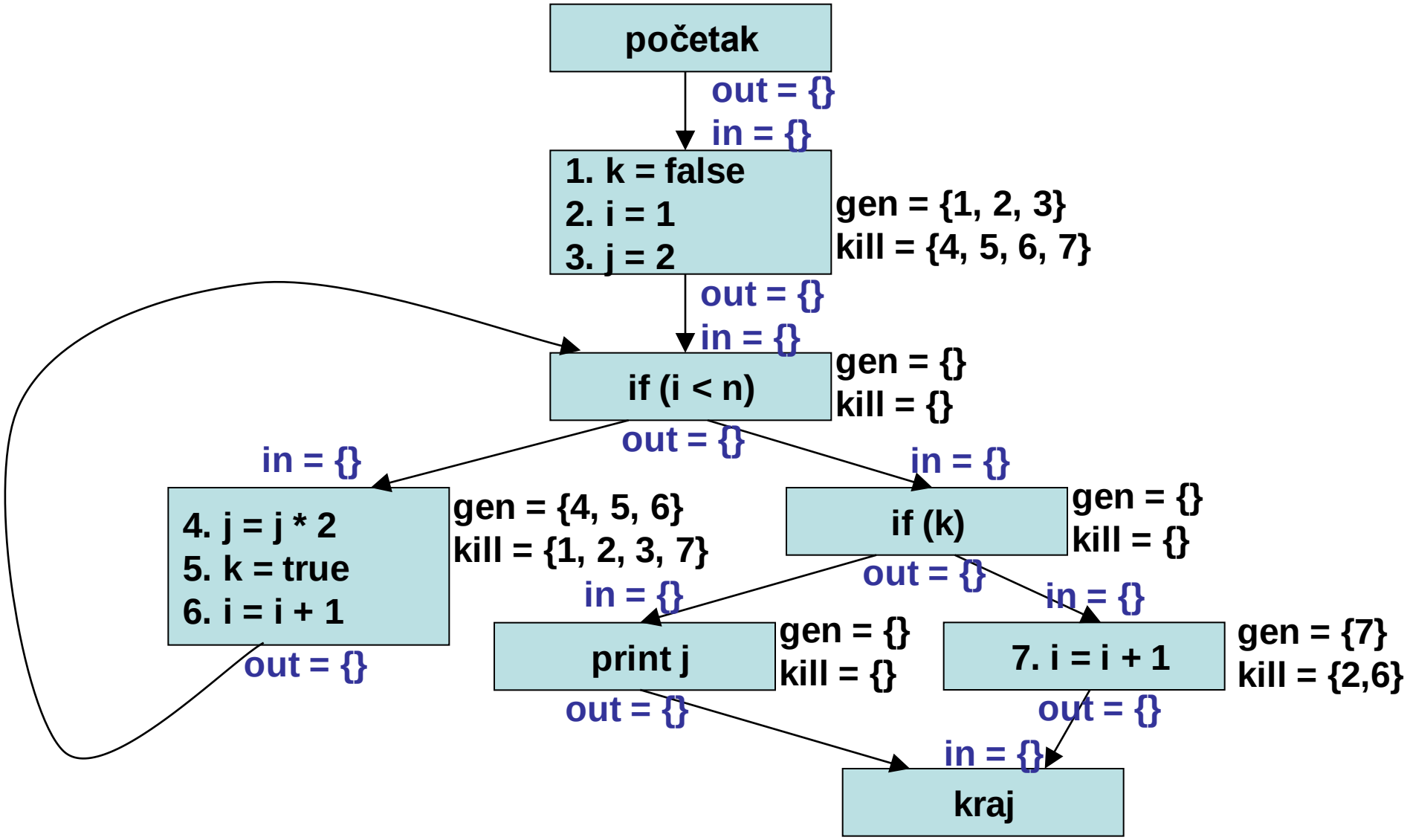
$$out[S] = gen[S] \cup (in[S] - kill[S])$$

$$in[S] = \bigcup_{P \text{ predecessor of } S} out[P]$$

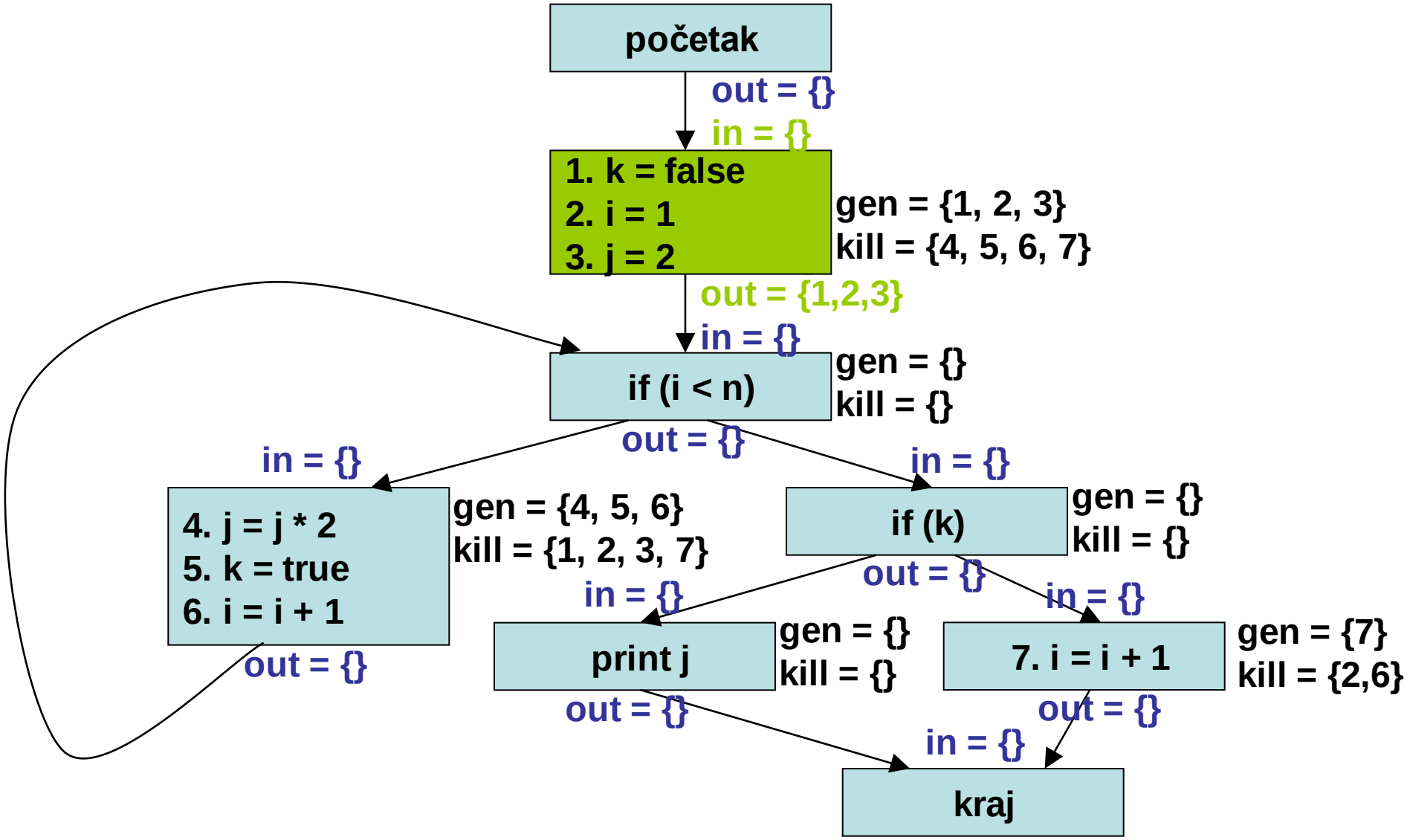
Primer


početak
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
if (i < n)
gen = {}
kill = {}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
if (k)
gen = {}
kill = {}
print j
gen = {}
kill = {}
7. i = i + 1
gen = {7}
kill = {2,6}
kraj

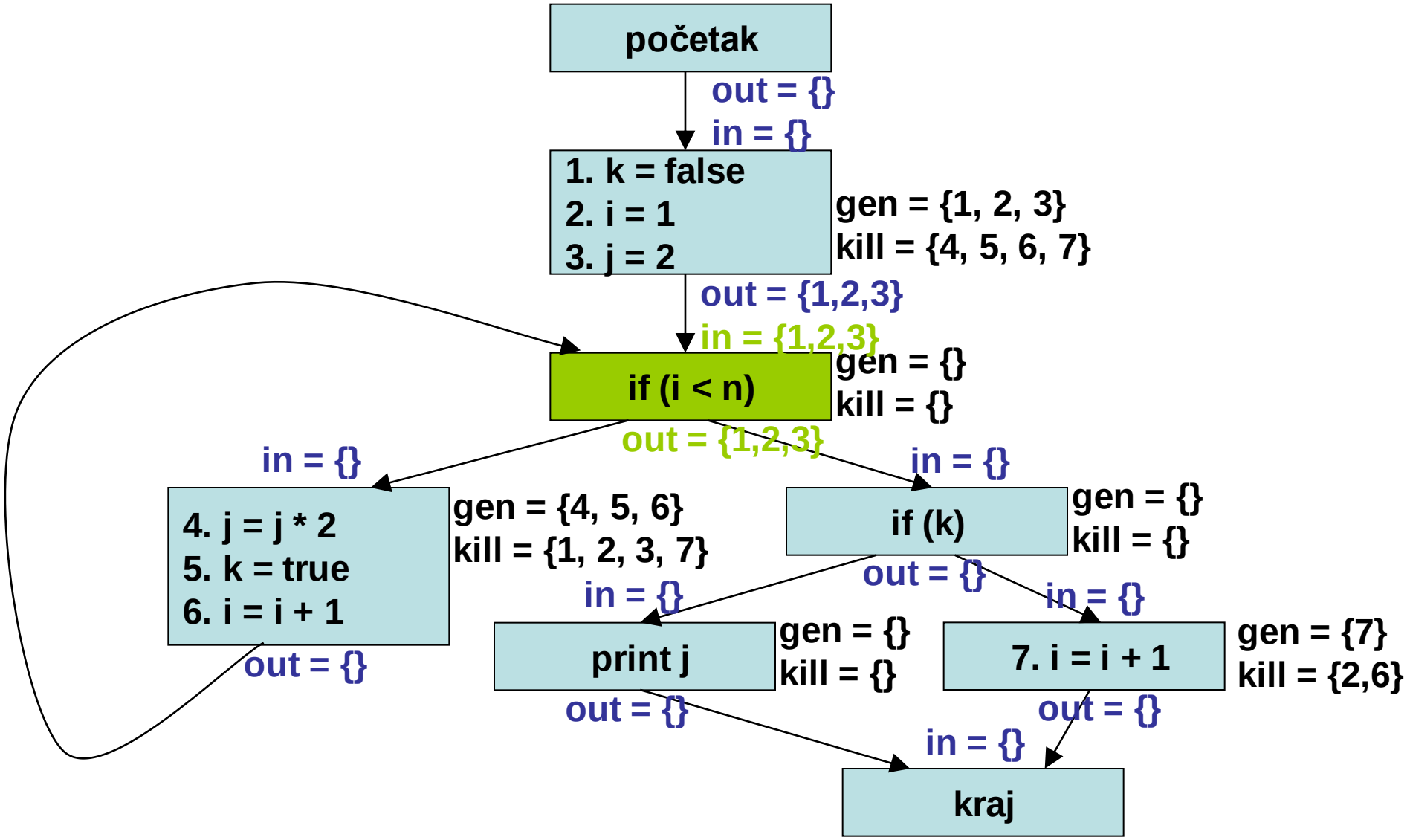
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {}
in = {}
if (i < n)
gen = {}
kill = {}
out = {}
in = {}
in = {}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {}
if (k)
gen = {}
kill = {}
out = {}
in = {}
in = {}
print j
gen = {}
kill = {}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {}
out = {}
in = {}
kraj

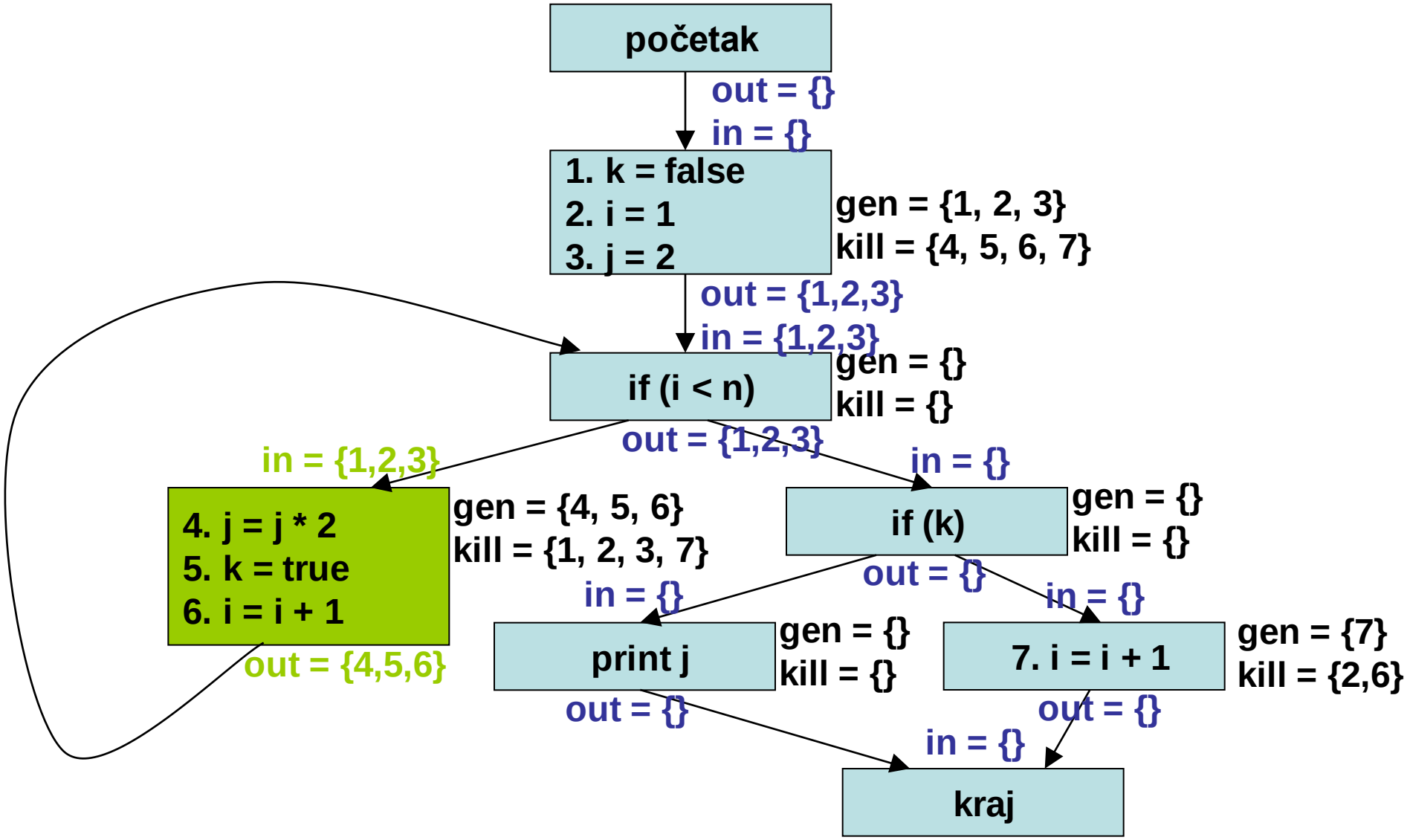
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {}
if (i < n)
gen = {}
kill = {}
out = {}
in = {}
in = {}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {}
if (k)
gen = {}
kill = {}
out = {}
in = {}
in = {}
print j
gen = {}
kill = {}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {}
out = {}
in = {}
kraj

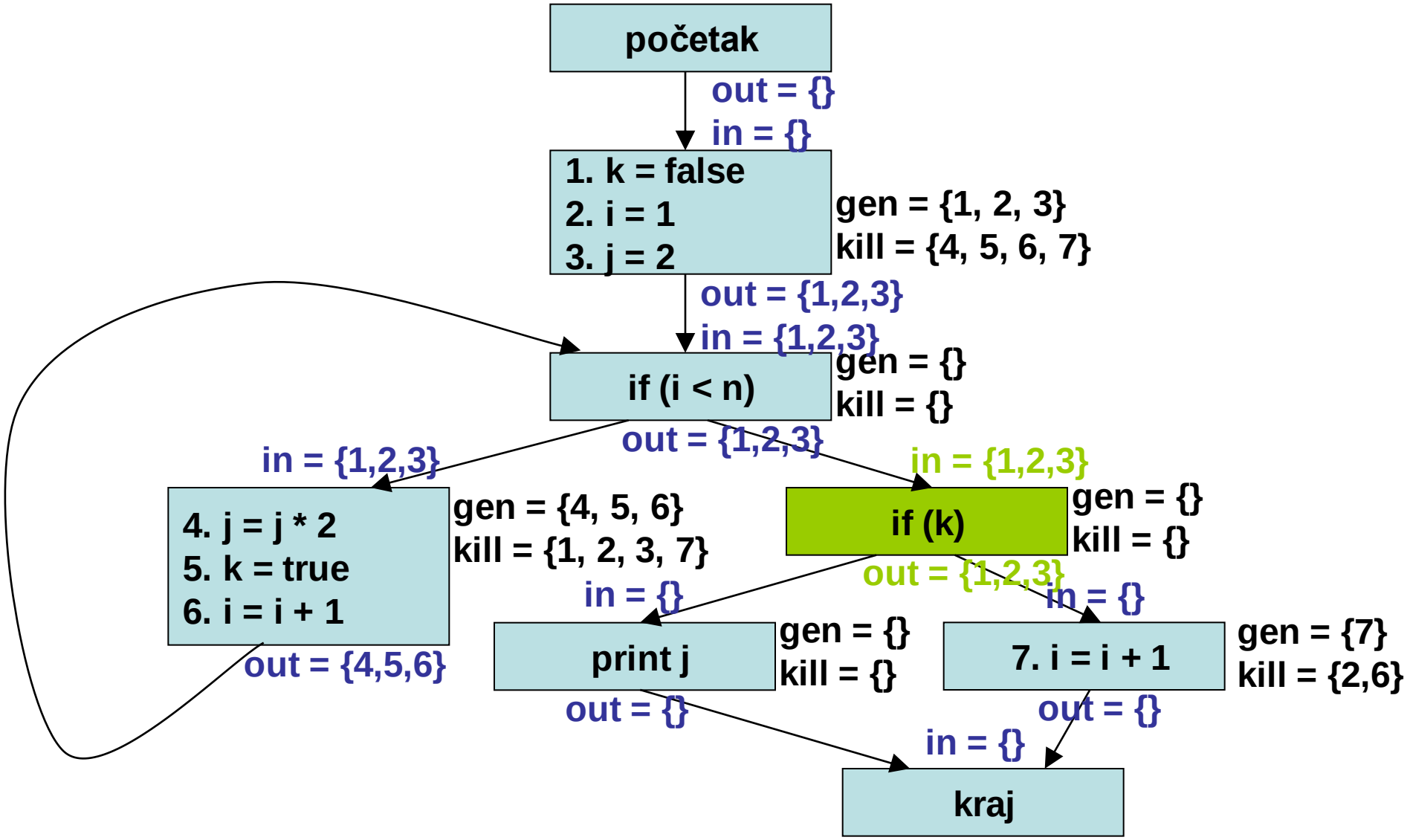
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3}
if (i < n)
gen = {}
kill = {}
out = {1,2,3}
in = {}
in = {}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {}
if (k)
gen = {}
kill = {}
out = {}
in = {}
in = {}
print j
gen = {}
kill = {}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {}
out = {}
in = {}
kraj

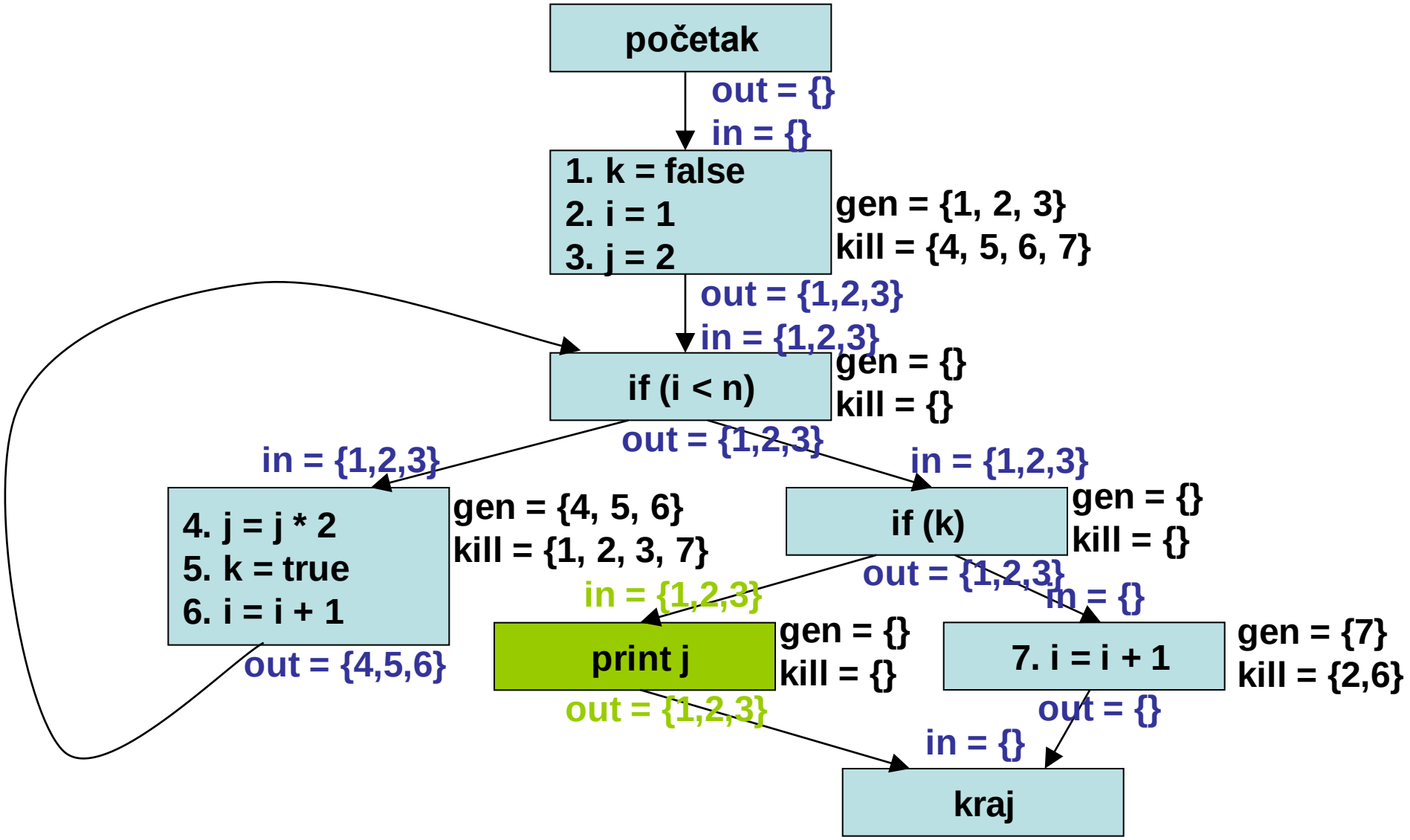
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3}
if (i < n)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
in = {}
if (k)
gen = {}
kill = {}
out = {}
in = {}
print j
gen = {}
kill = {}
out = {}
in = {}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {}
in = {}
kraj

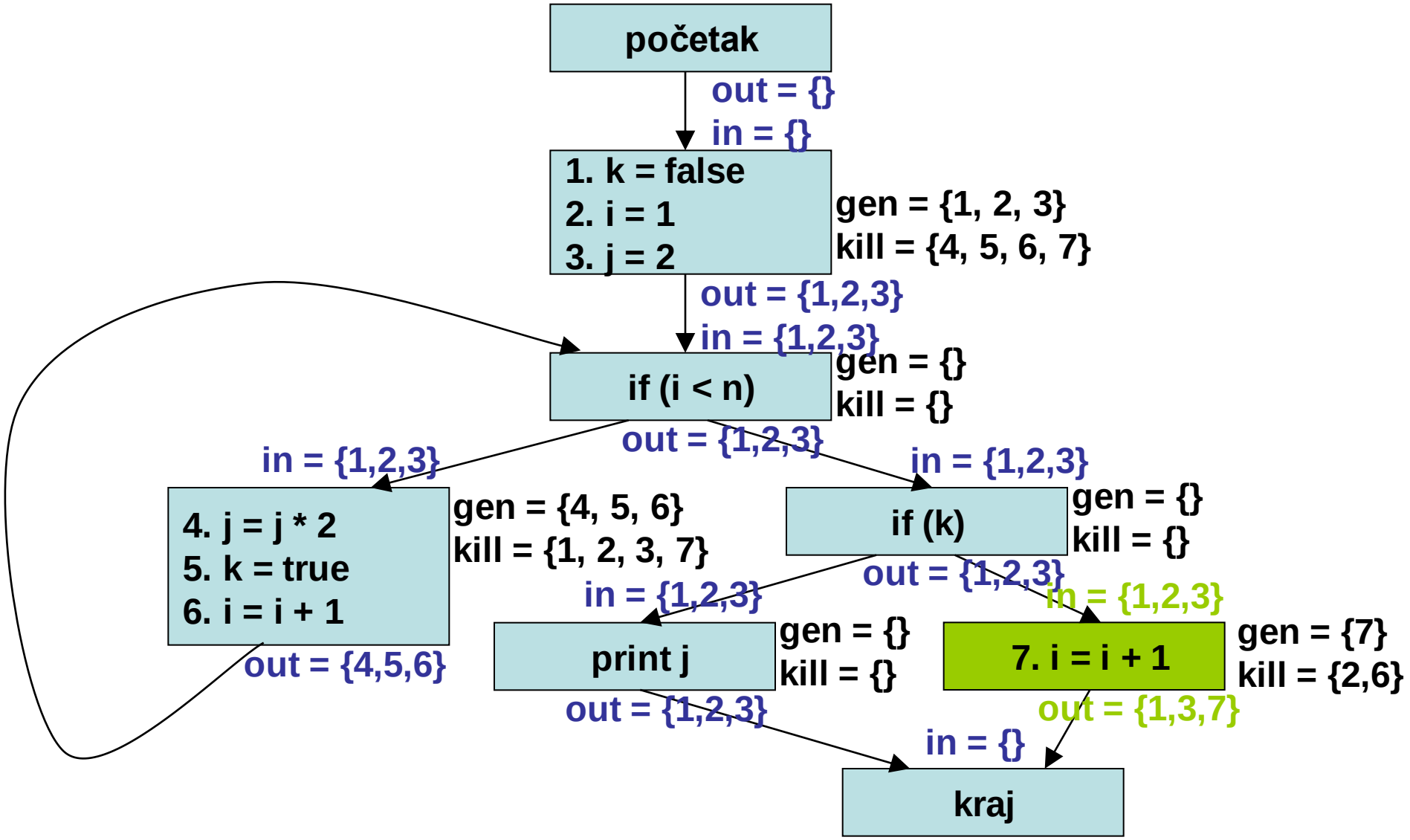
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3}
if (i < n)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3}
in = {}
in = {}
print j
gen = {}
kill = {}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {}
out = {}
in = {}
kraj

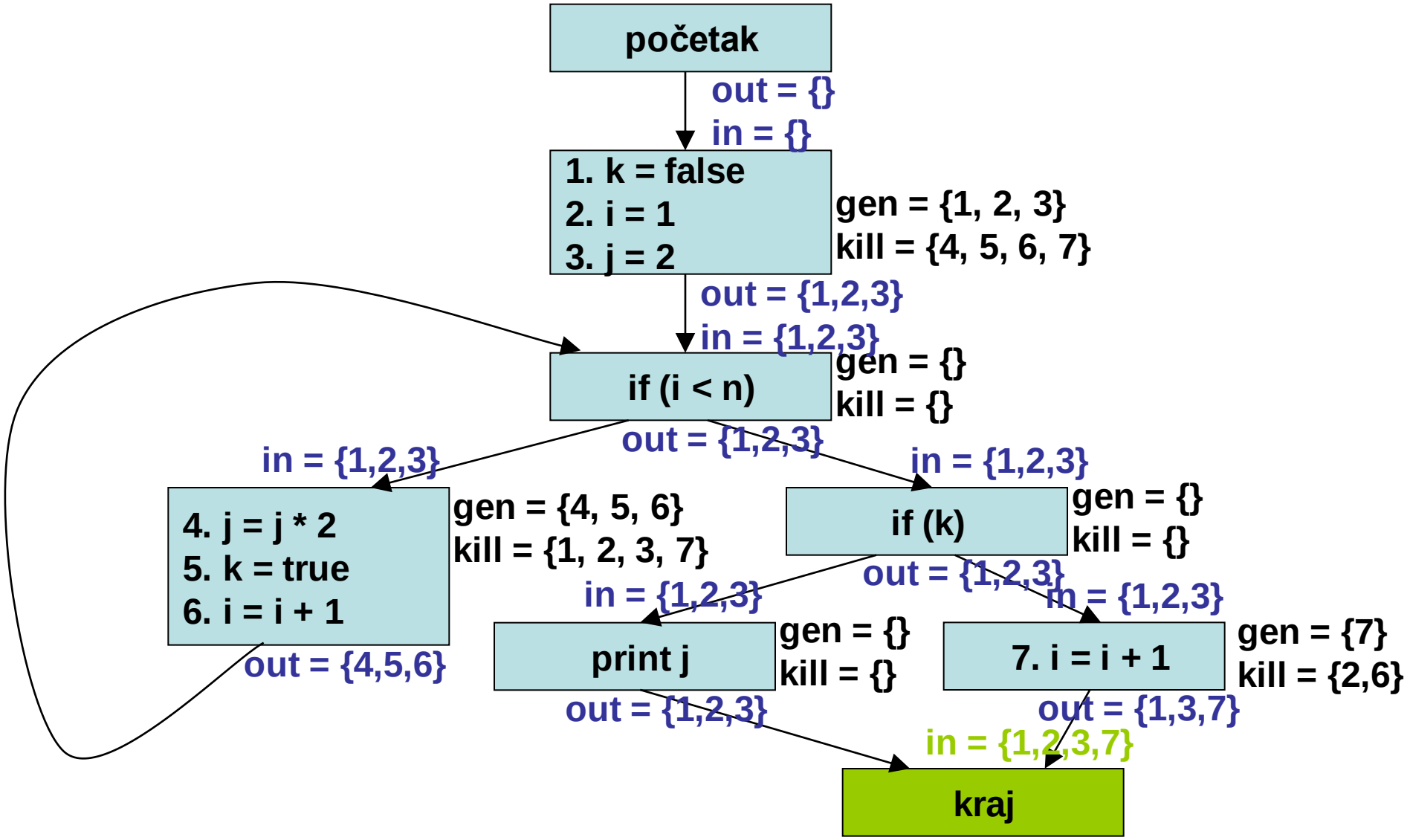
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3}
if (i < n)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {}
print j
gen = {}
kill = {}
out = {1,2,3}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {}
in = {}
kraj

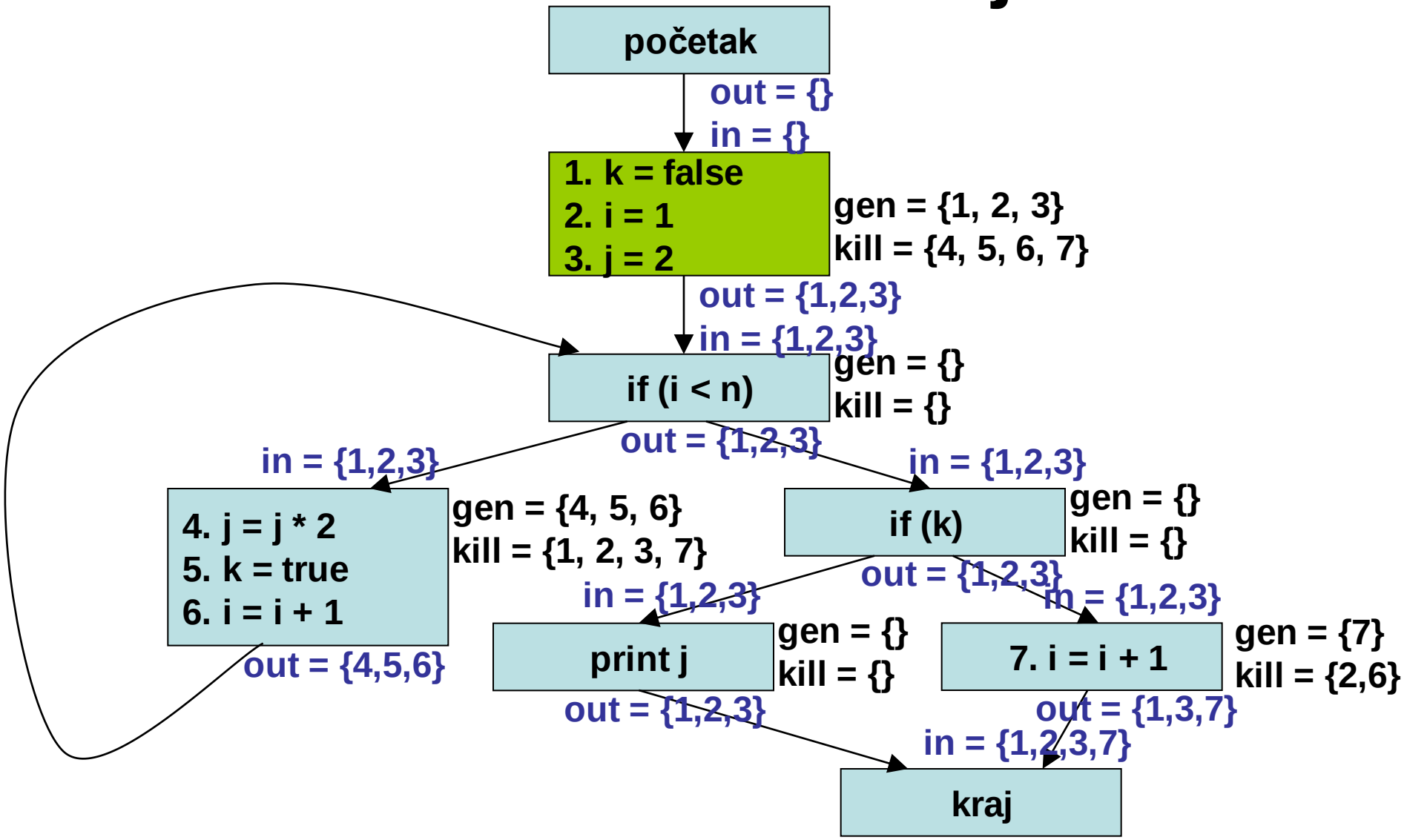
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3}
if (i < n)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
print j
gen = {}
kill = {}
out = {1,2,3}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,7}
in = {}
kraj

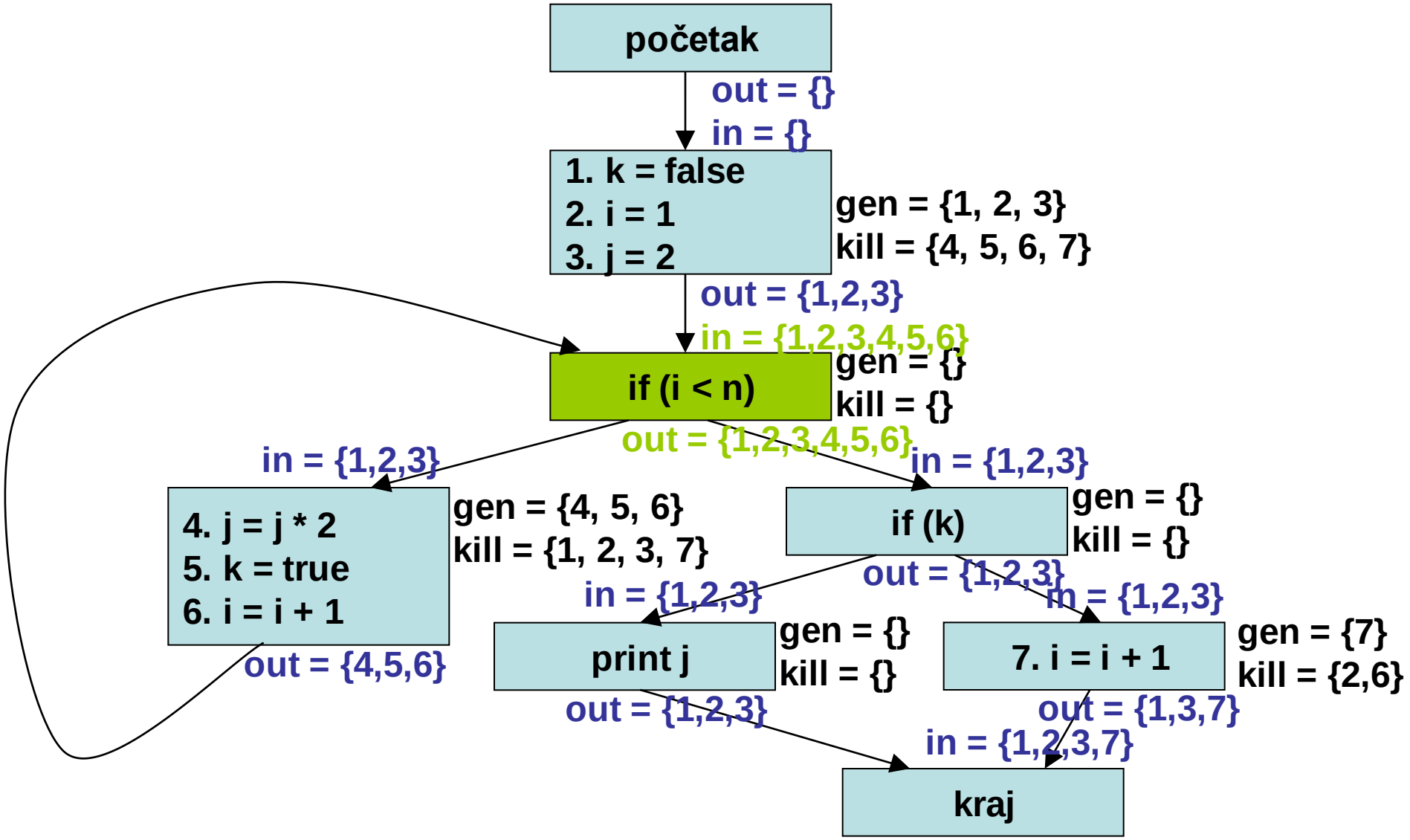
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3}
if (i < n)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
print j
gen = {}
kill = {}
out = {1,2,3}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,7}
in = {1,2,3,7}
kraj

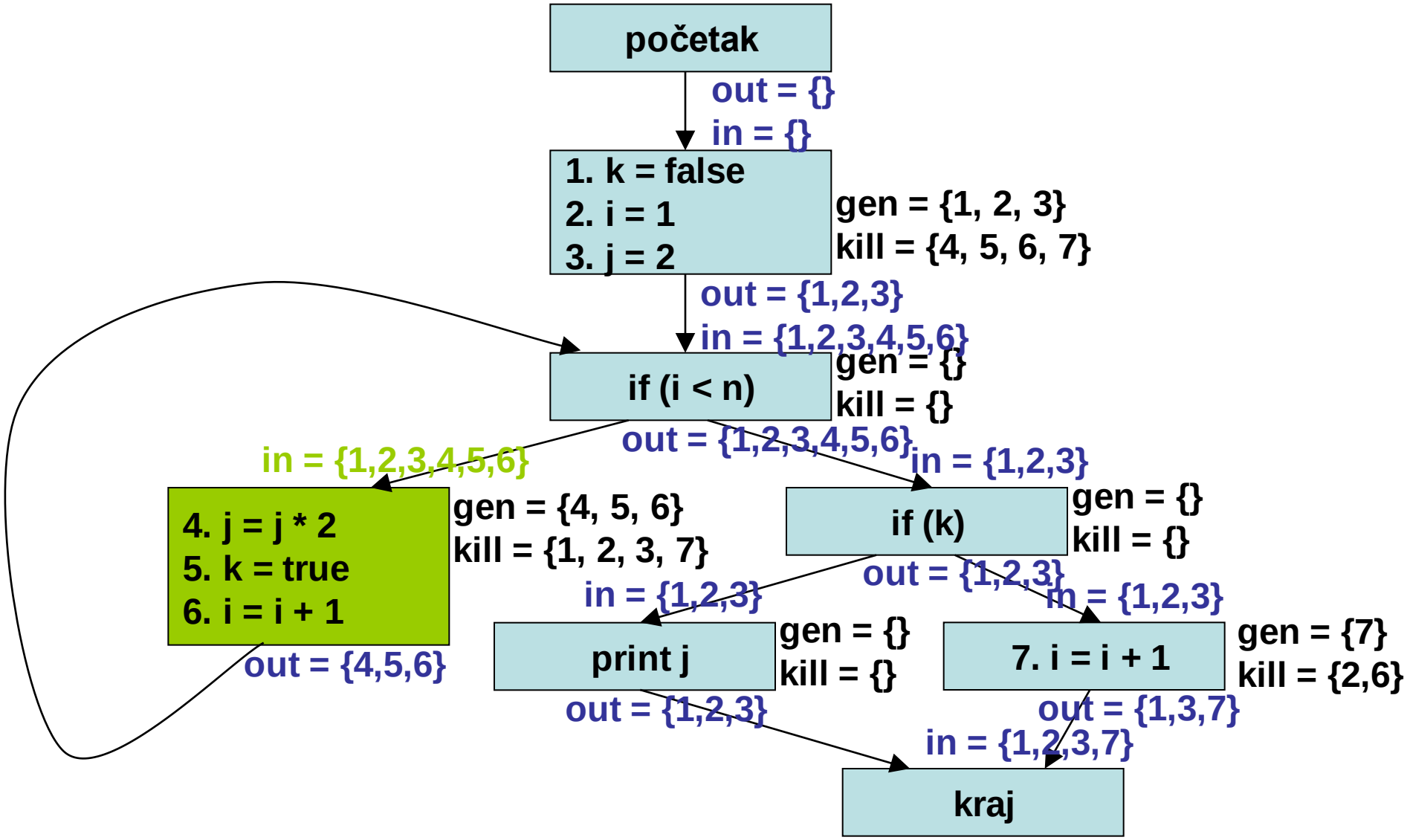
Primer – 2. iteracija



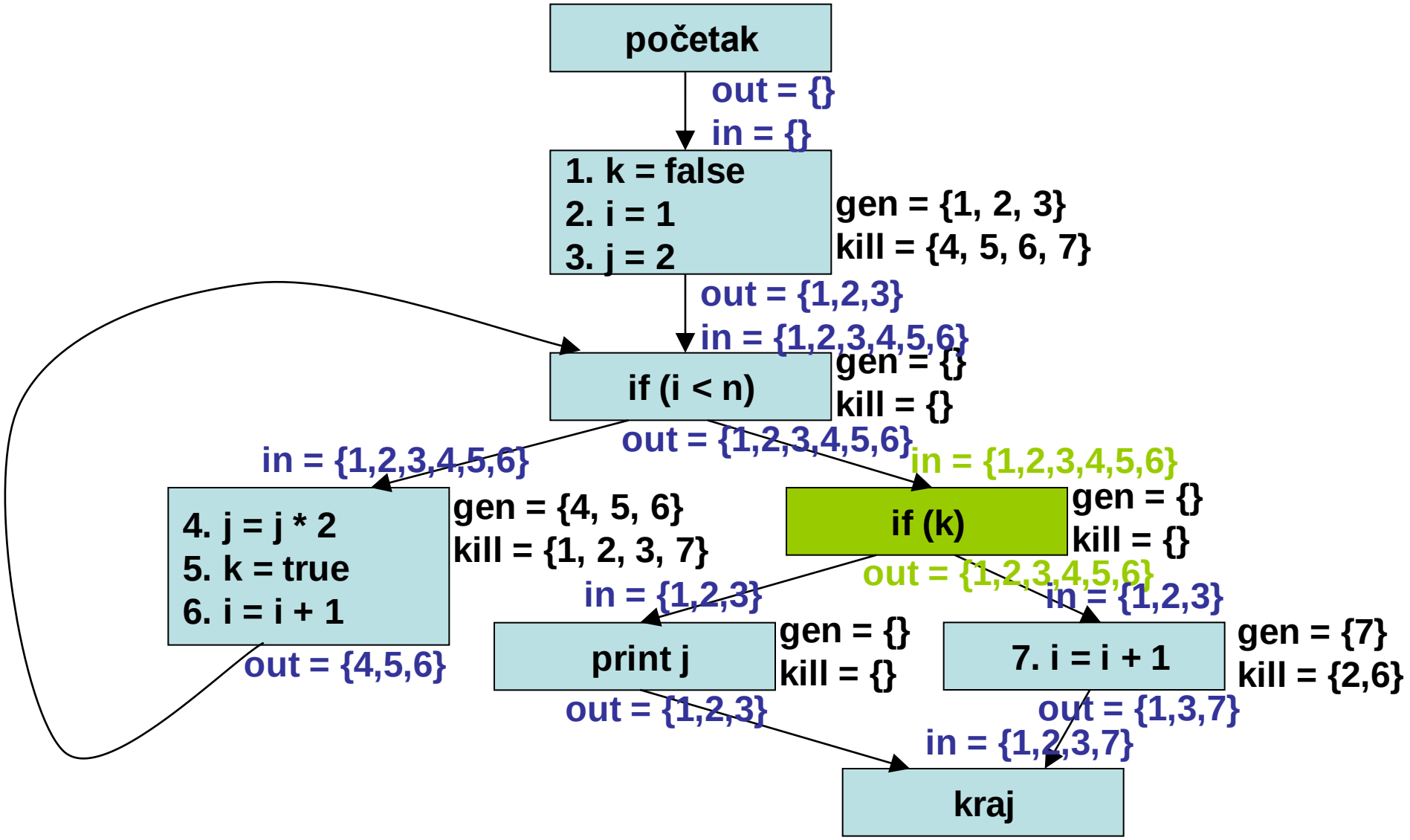
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
print j
gen = {}
kill = {}
out = {1,2,3}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,7}
in = {1,2,3,7}
kraj

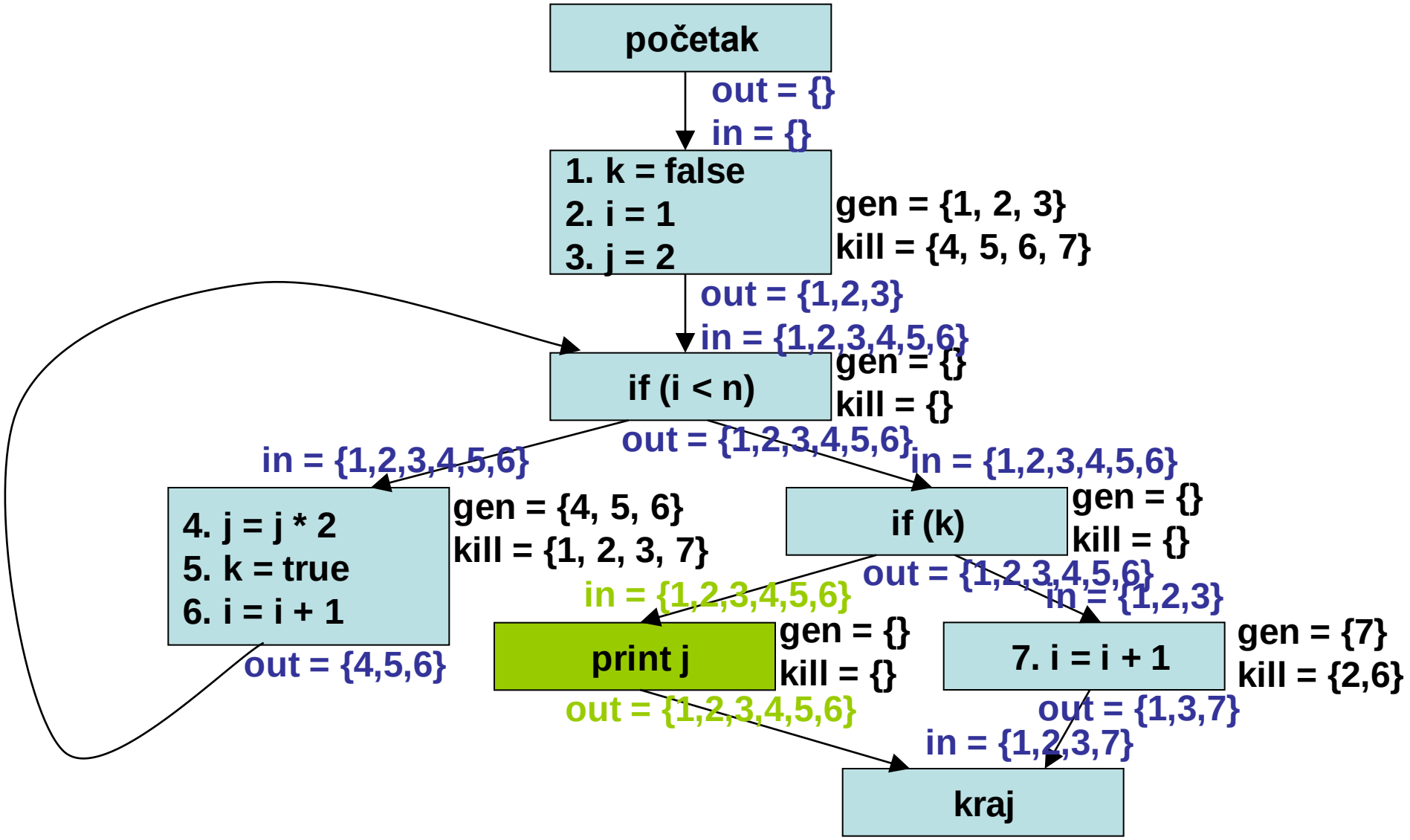
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3}
in = {1,2,3}
in = {1,2,3}
print j
gen = {}
kill = {}
out = {1,2,3}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,7}
in = {1,2,3,7}
kraj

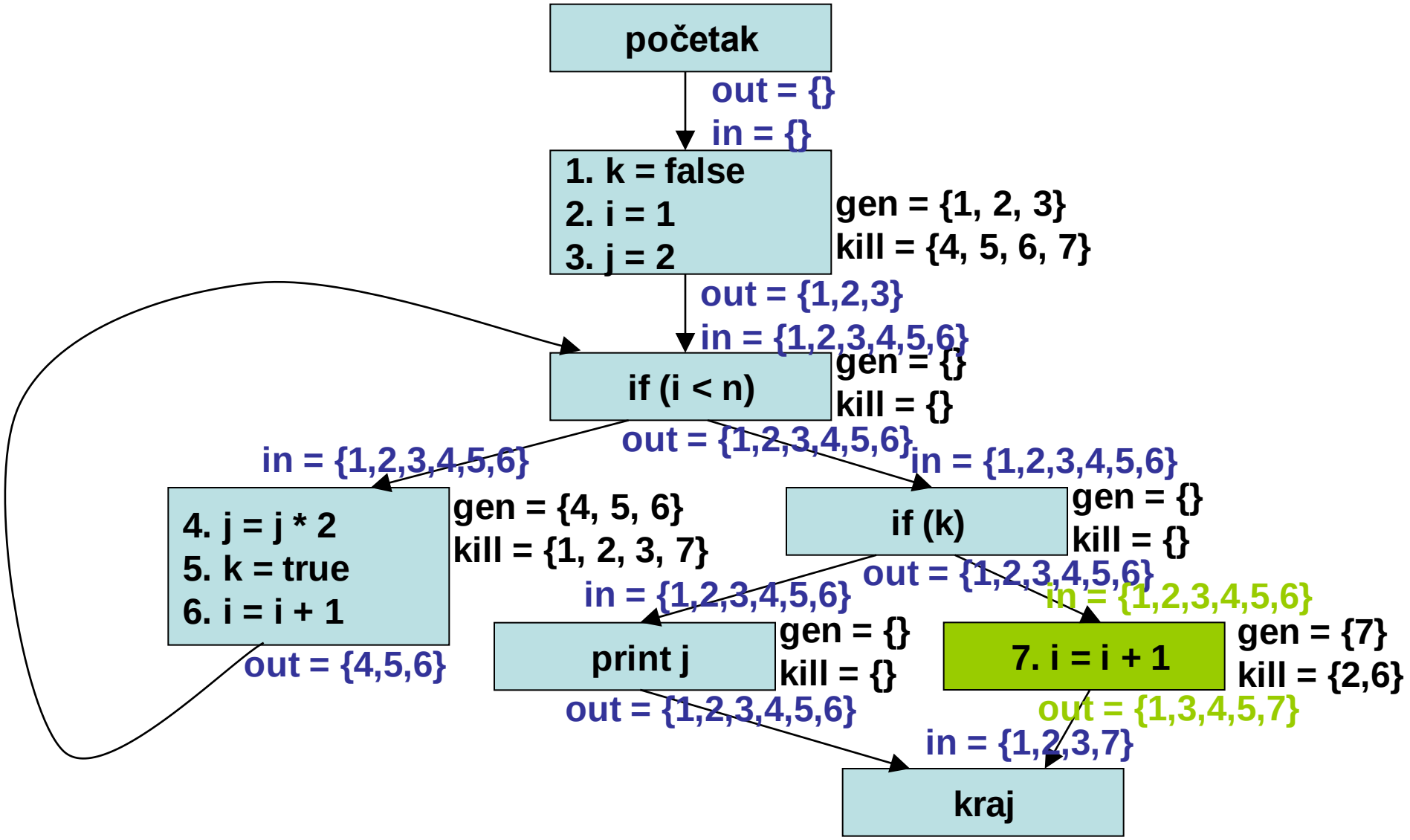
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3}
in = {1,2,3}
print j
gen = {}
kill = {}
out = {1,2,3}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,7}
in = {1,2,3,7}
kraj

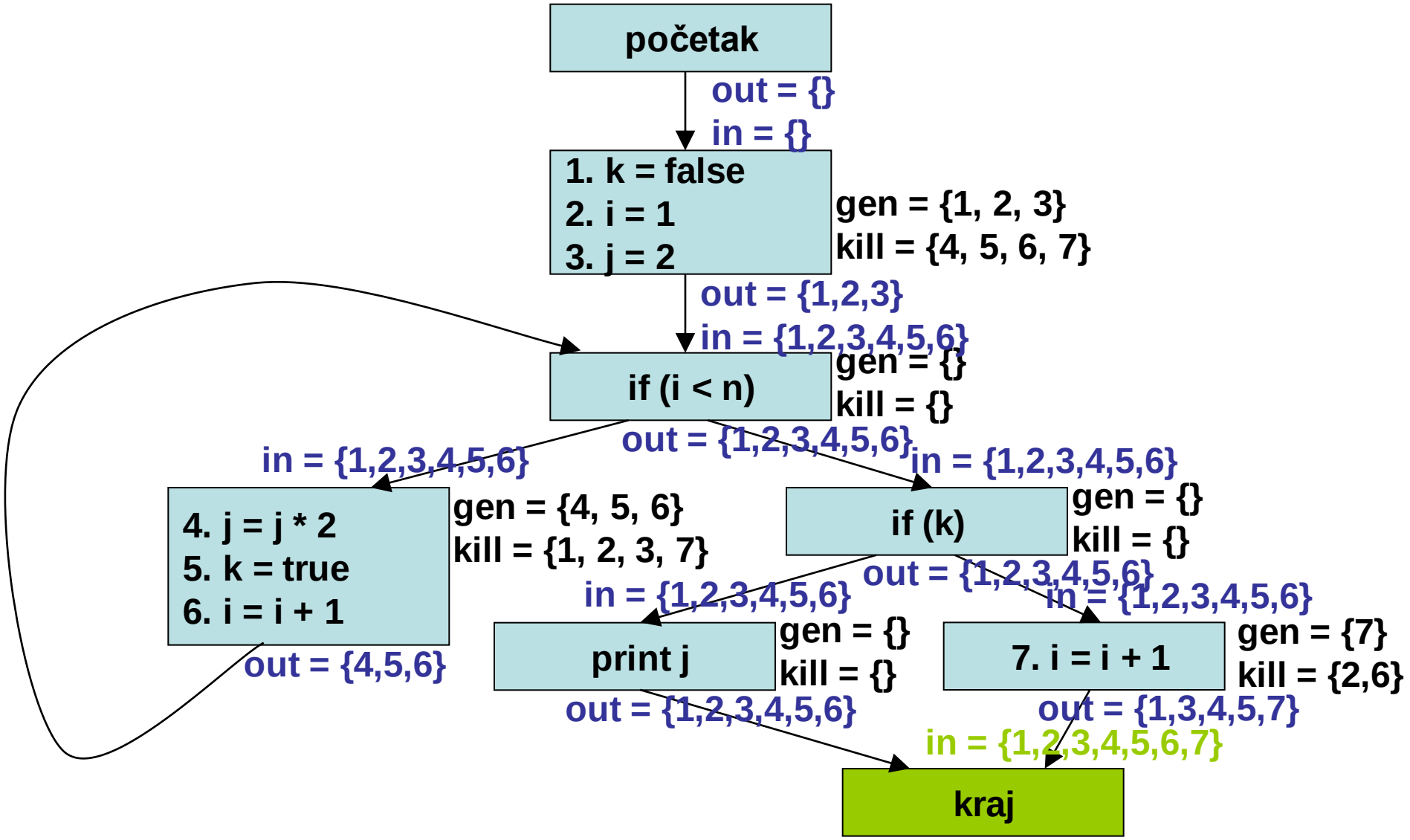
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,7}
in = {1,2,3,7}
kraj

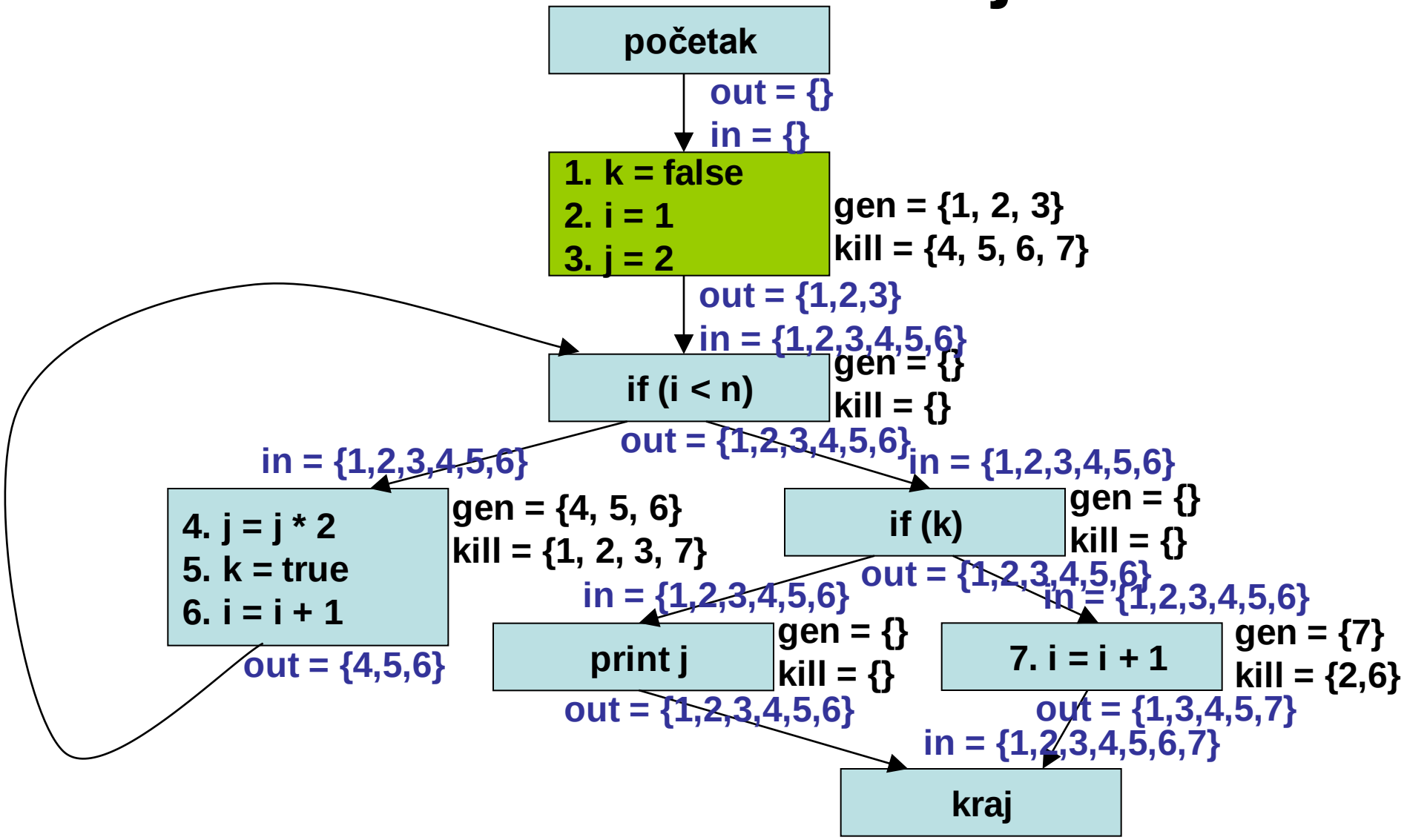
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,7}
kraj

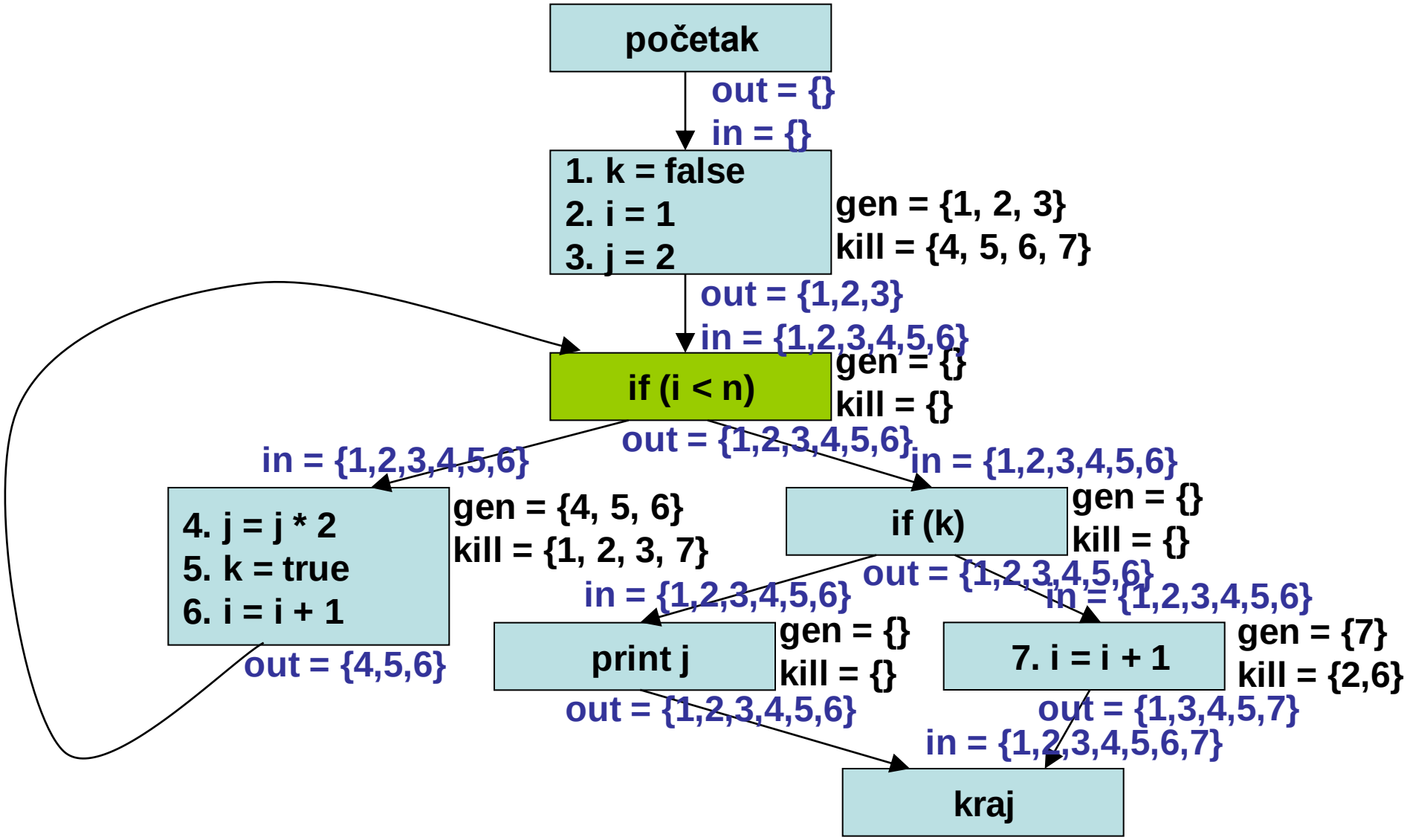
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

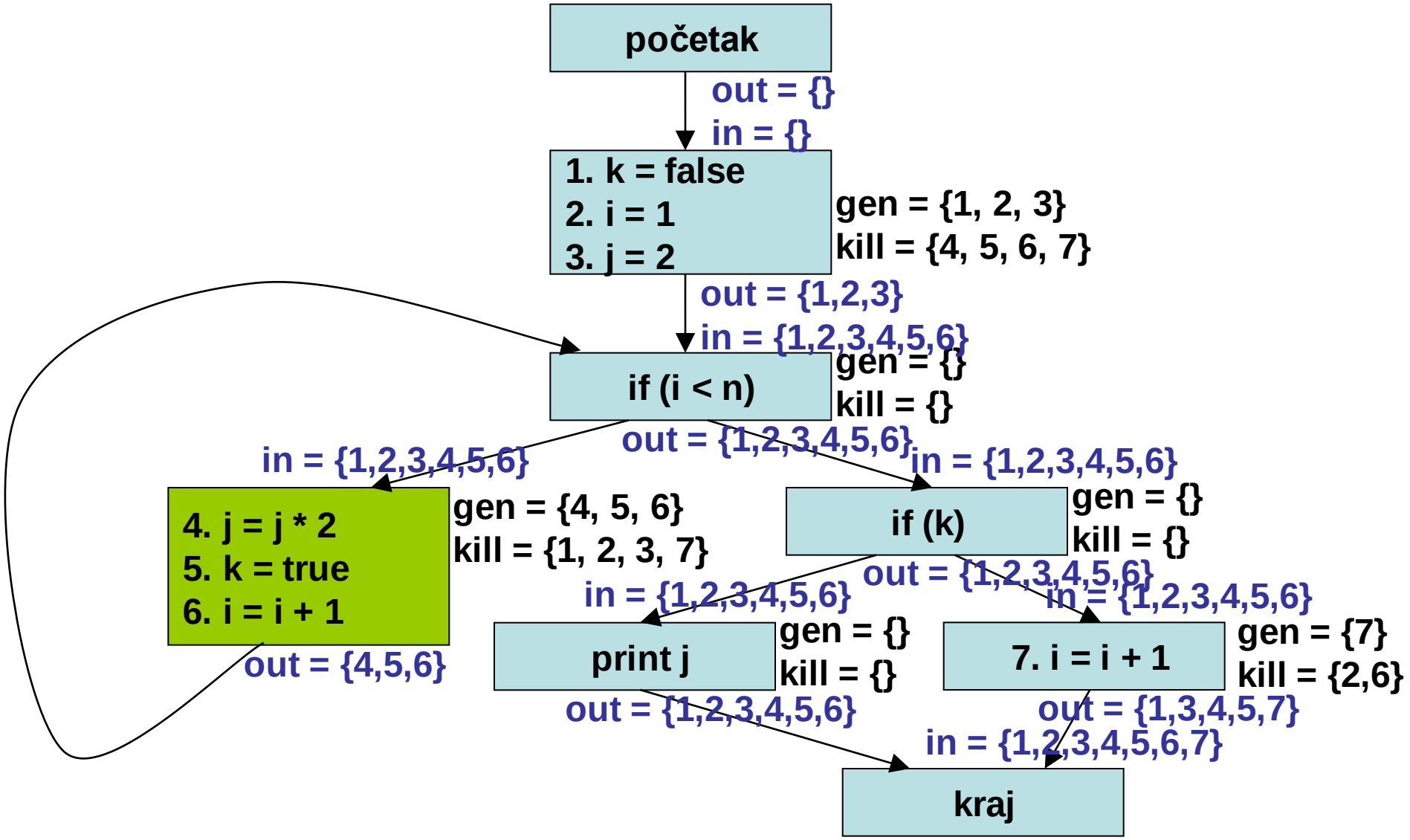
Primer – 3. iteracija



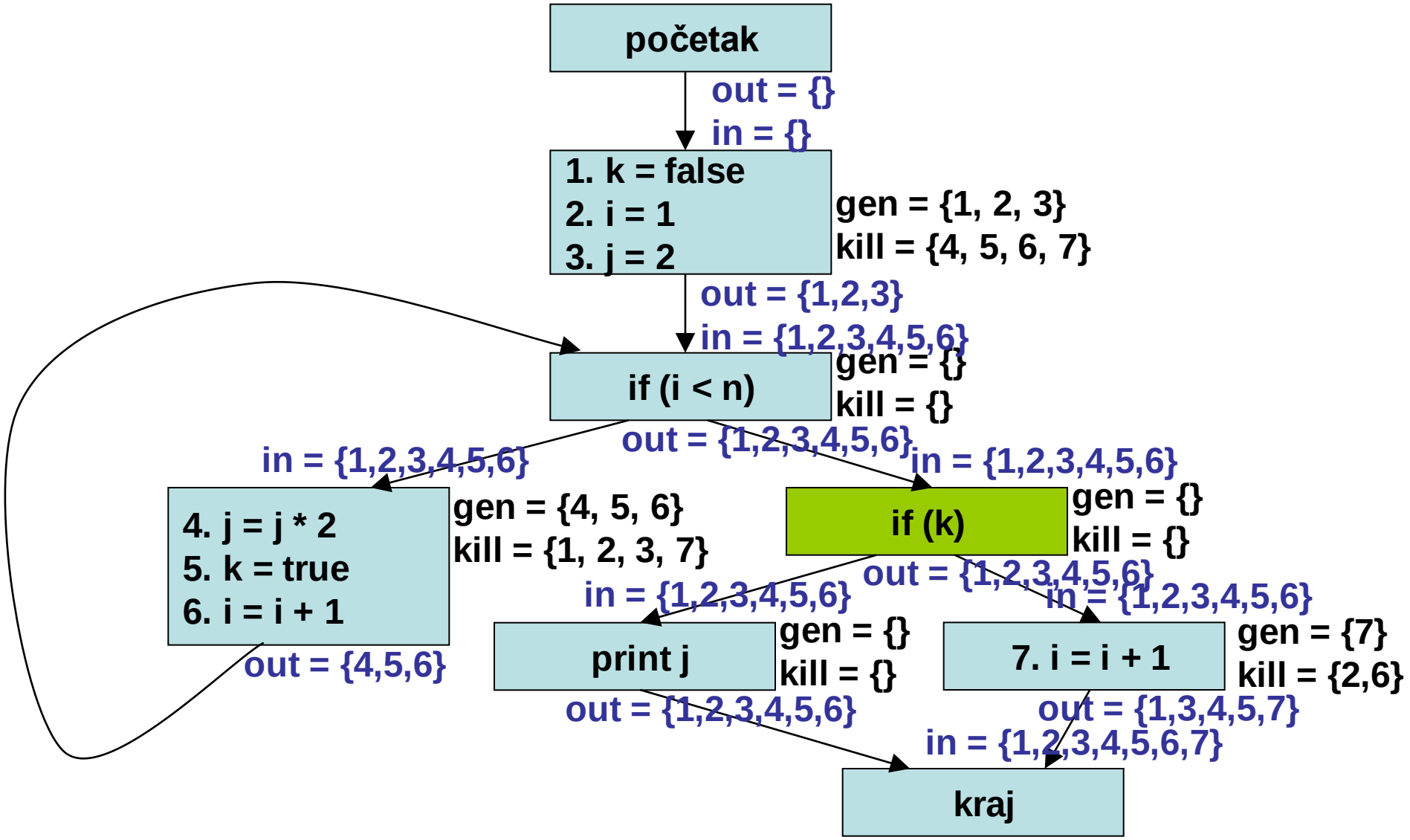
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

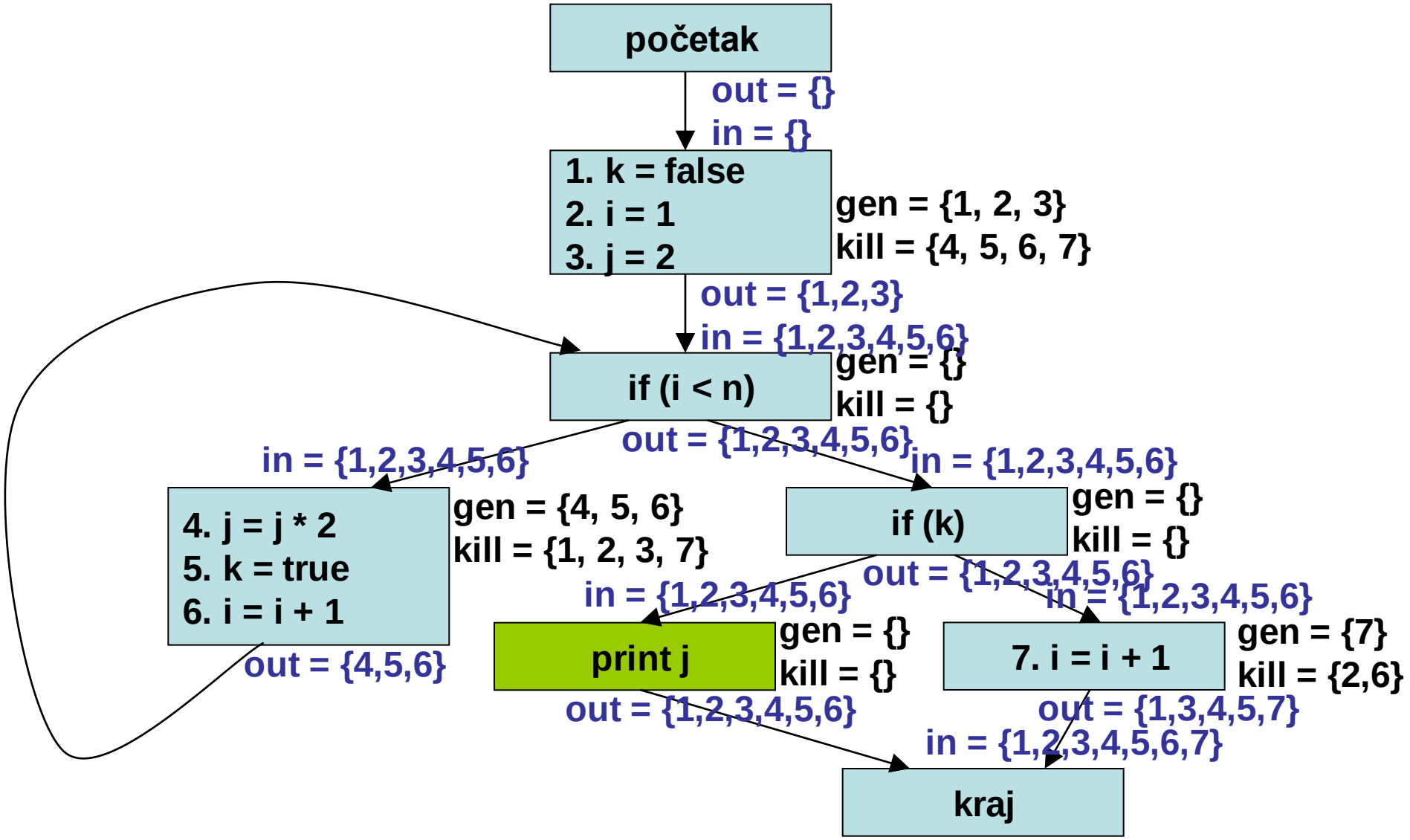
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

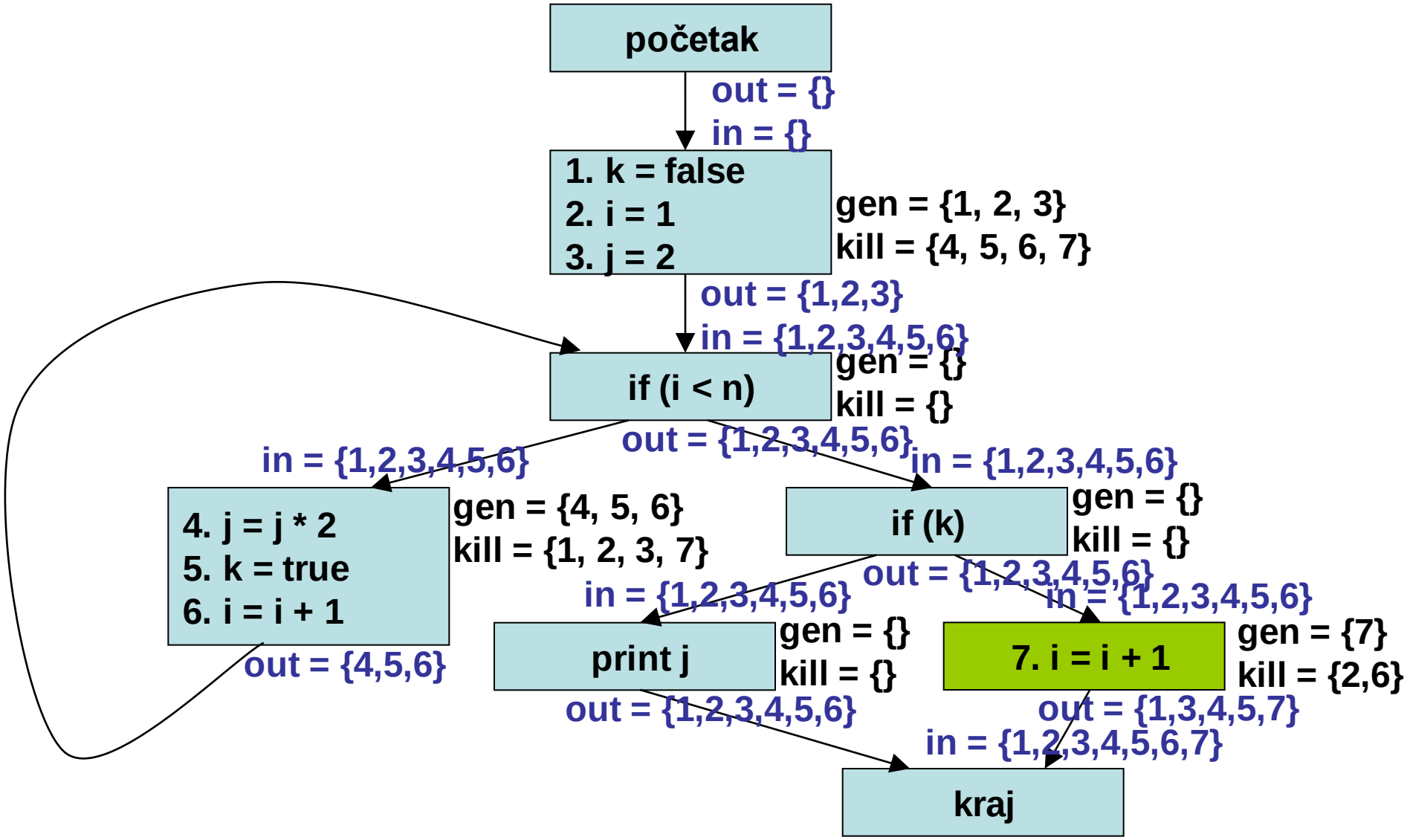
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

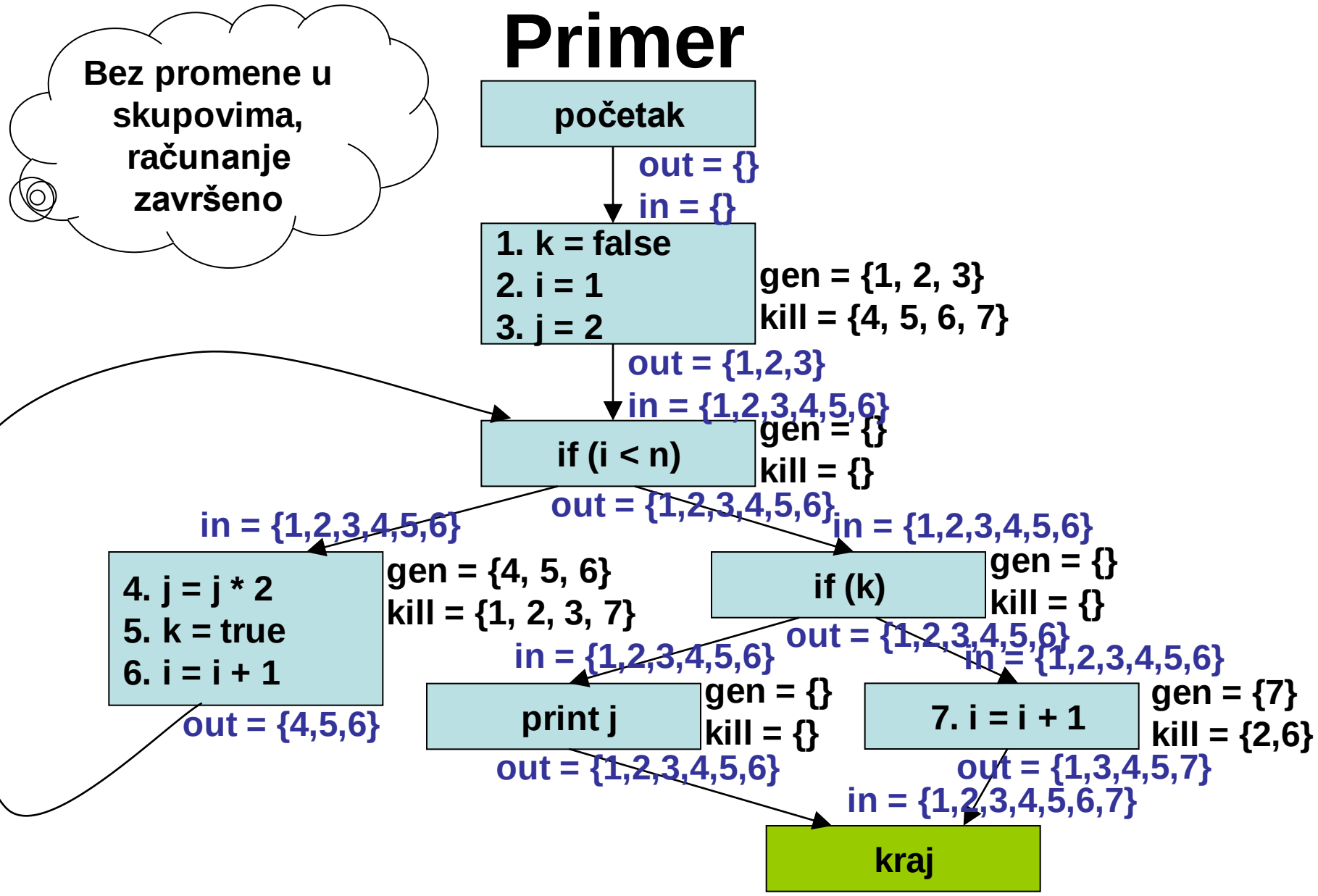
Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

Primer


početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

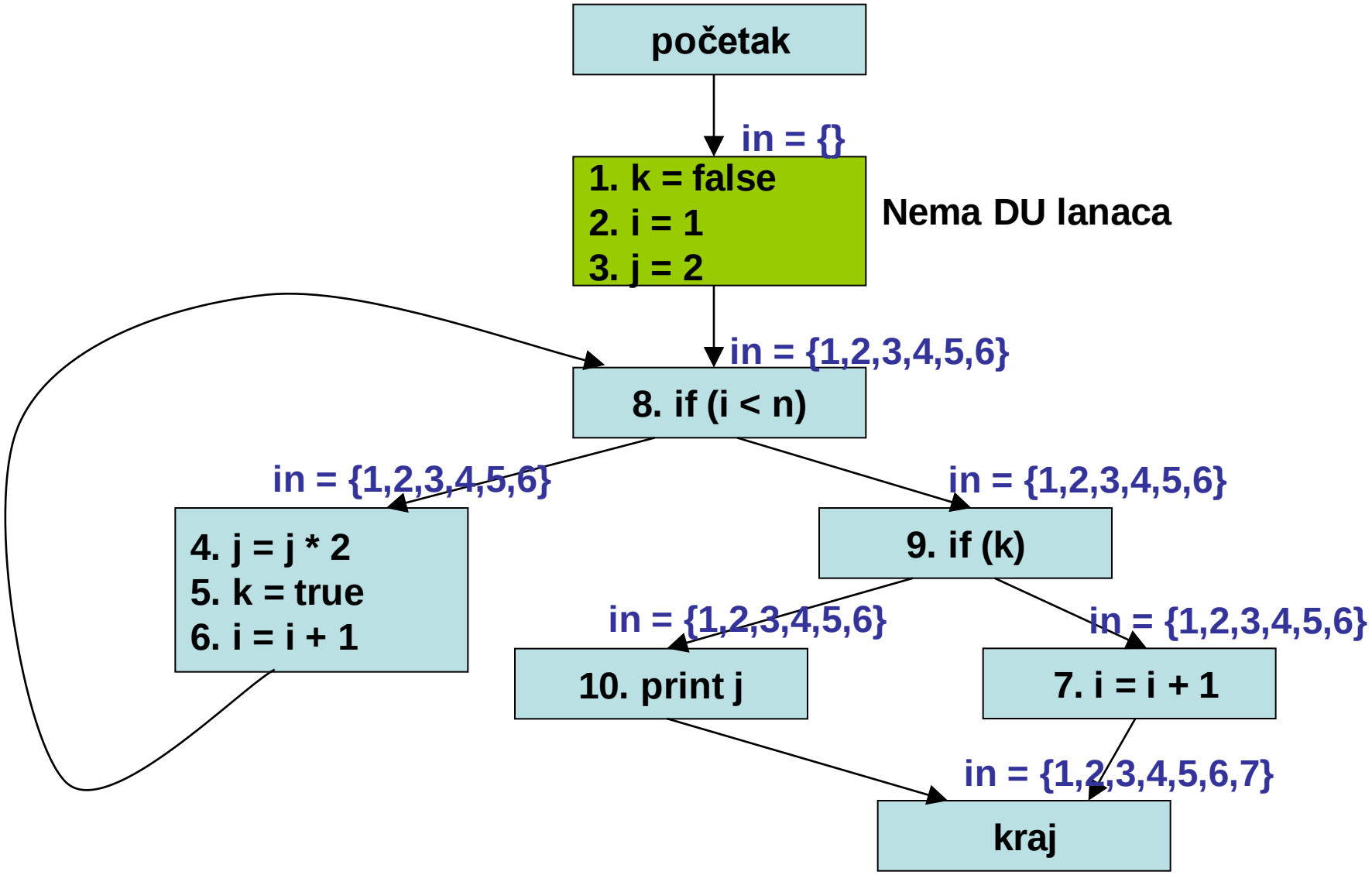
Primer


Bez promene u skupovima, računanje završeno
početak
out = {}
in = {}
1. k = false
2. i = 1
3. j = 2
gen = {1, 2, 3}
kill = {4, 5, 6, 7}
out = {1,2,3}
in = {1,2,3,4,5,6}
if (i < n)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
gen = {4, 5, 6}
kill = {1, 2, 3, 7}
out = {4,5,6}
if (k)
gen = {}
kill = {}
out = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
print j
gen = {}
kill = {}
out = {1,2,3,4,5,6}
7. i = i + 1
gen = {7}
kill = {2,6}
out = {1,3,4,5,7}
in = {1,2,3,4,5,6,7}
kraj

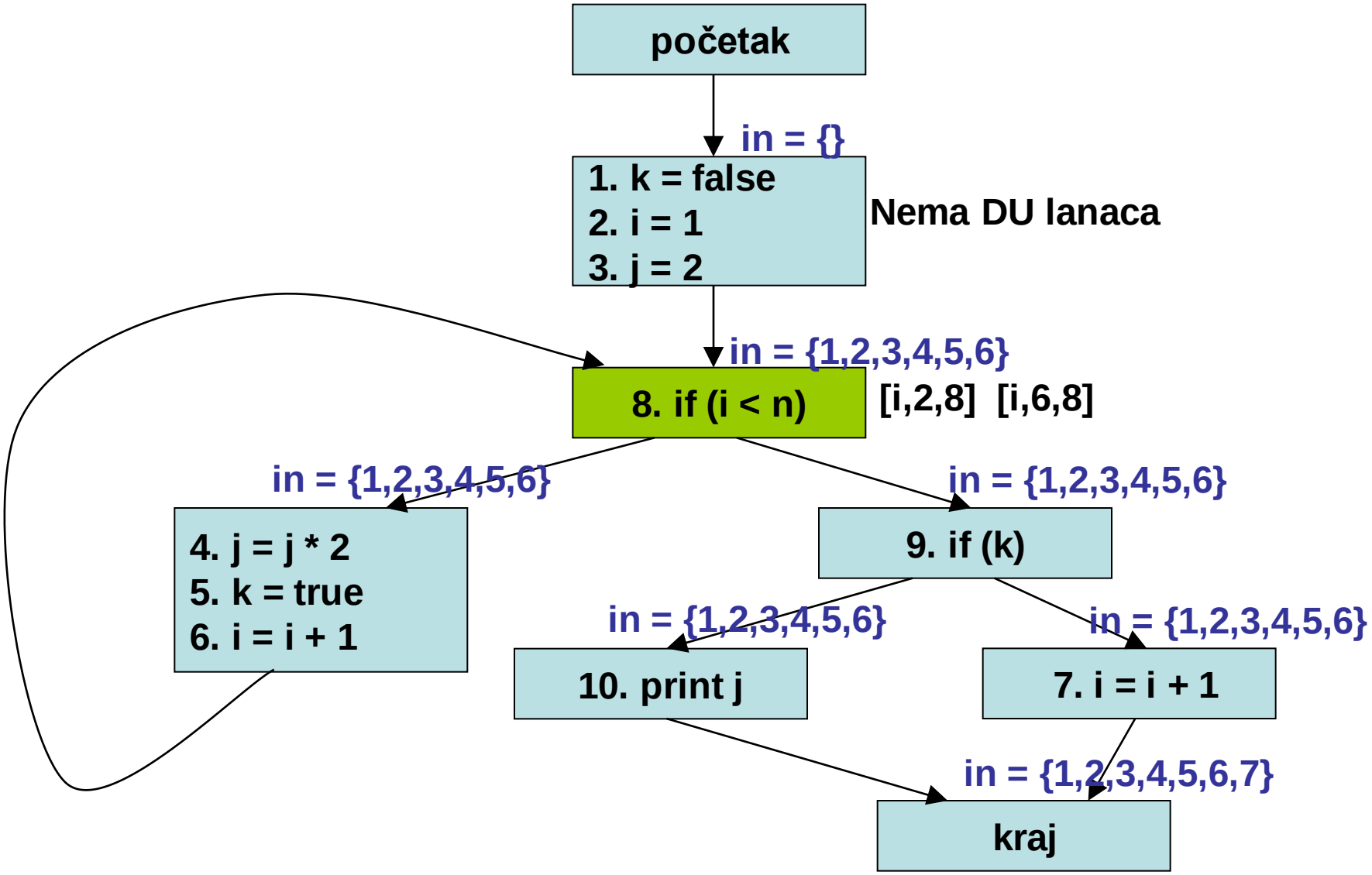
Računanje DU lanaca

- U bloku *S*:
  - DU lanac je određen definicijom x iz *in*[*S*] i lokalnom upotrebom x
    - Napomena: Lokalna definicija x koja prethodi upotrebi x "ubija" definicije x iz in skupa i ranije definicije x iz S

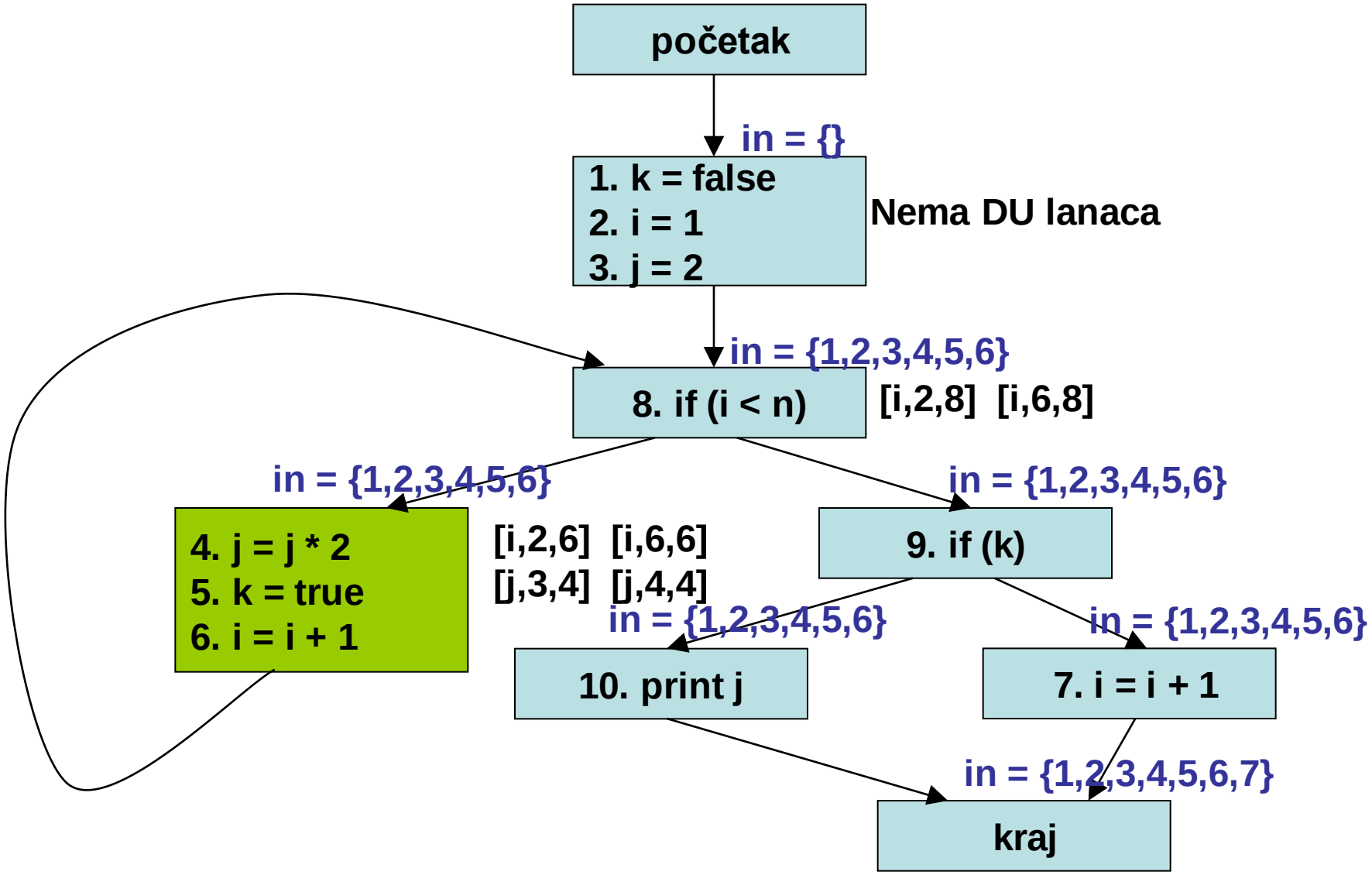
Primer


početak
in = {}
1. k = false
2. i = 1
3. j = 2
Nema DU lanaca
in = {1,2,3,4,5,6}
8. if (i < n)
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
9. if (k)
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
10. print j
7. i = i + 1
in = {1,2,3,4,5,6,7}
kraj

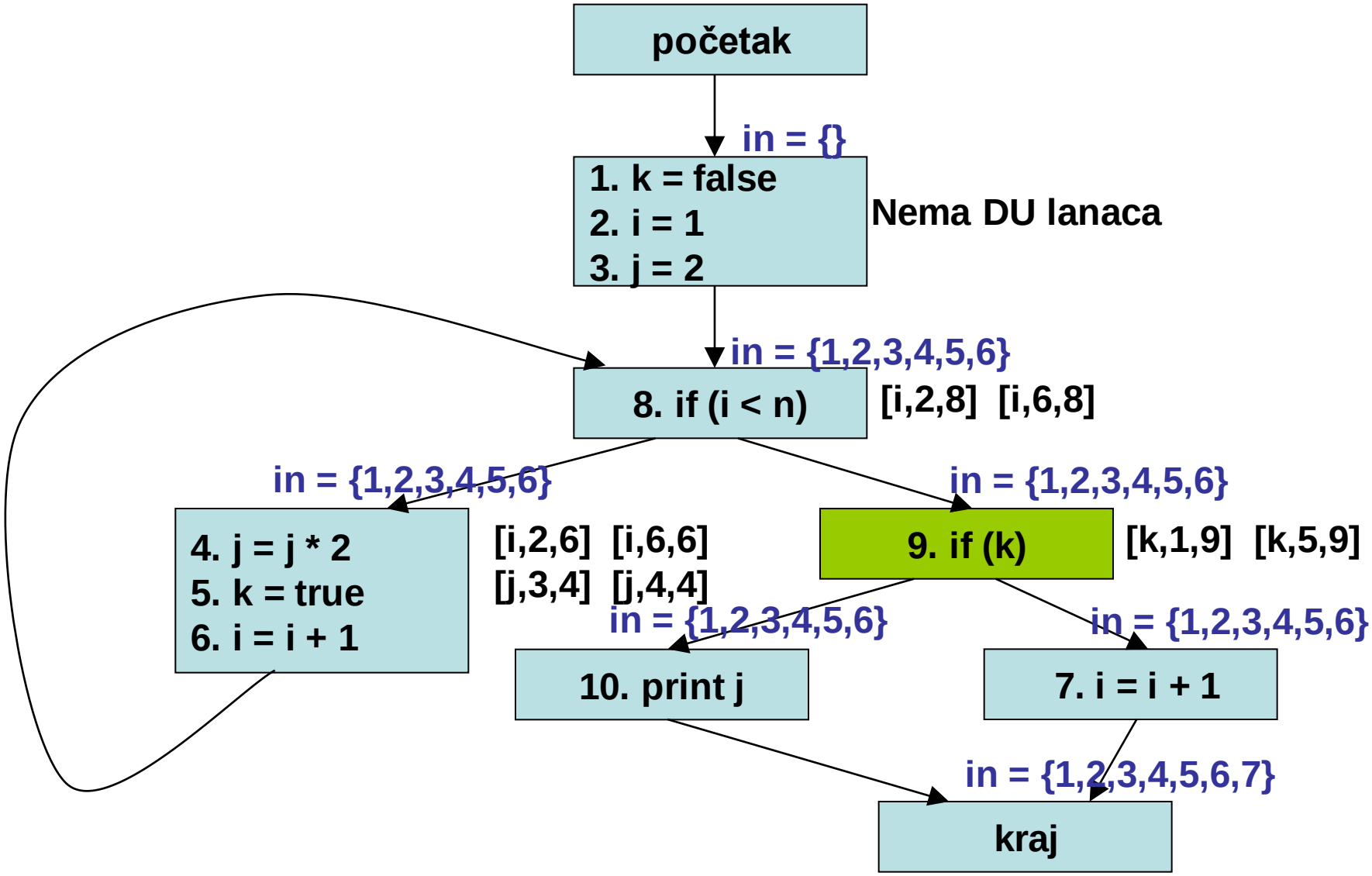
Primer


početak
in = {}
1. k = false
2. i = 1
3. j = 2
Nema DU lanaca
in = {1,2,3,4,5,6}
8. if (i < n)
[i,2,8] [i,6,8]
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
in = {1,2,3,4,5,6}
9. if (k)
in = {1,2,3,4,5,6}
10. print j
in = {1,2,3,4,5,6}
7. i = i + 1
in = {1,2,3,4,5,6,7}
kraj

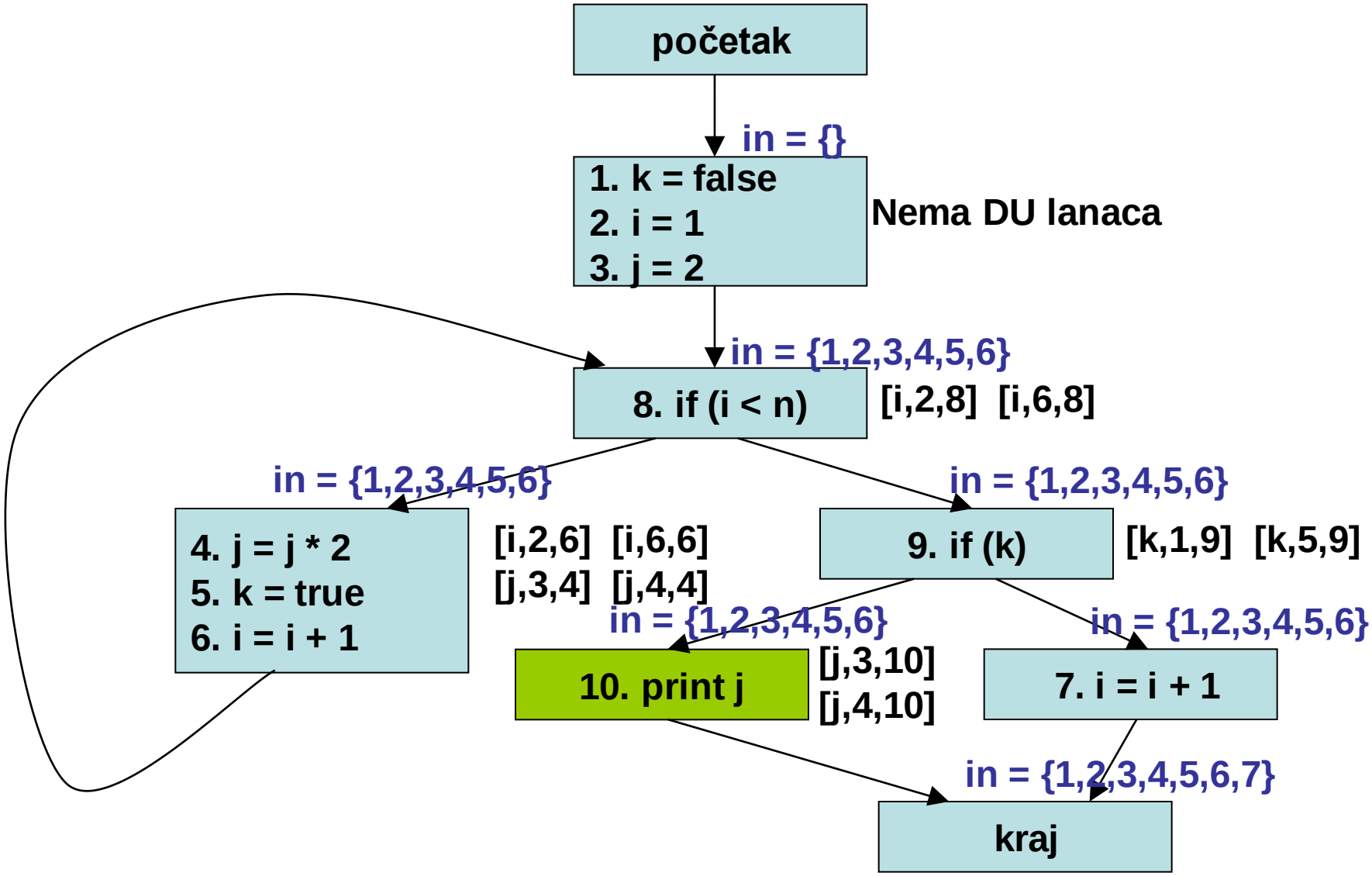
Primer


početak
in = {}
1. k = false
2. i = 1
3. j = 2
Nema DU lanaca
in = {1,2,3,4,5,6}
8. if (i < n)
[i,2,8] [i,6,8]
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
[i,2,6] [i,6,6]
[j,3,4] [j,4,4]
9. if (k)
in = {1,2,3,4,5,6}
in = {1,2,3,4,5,6}
10. print j
7. i = i + 1
in = {1,2,3,4,5,6,7}
kraj

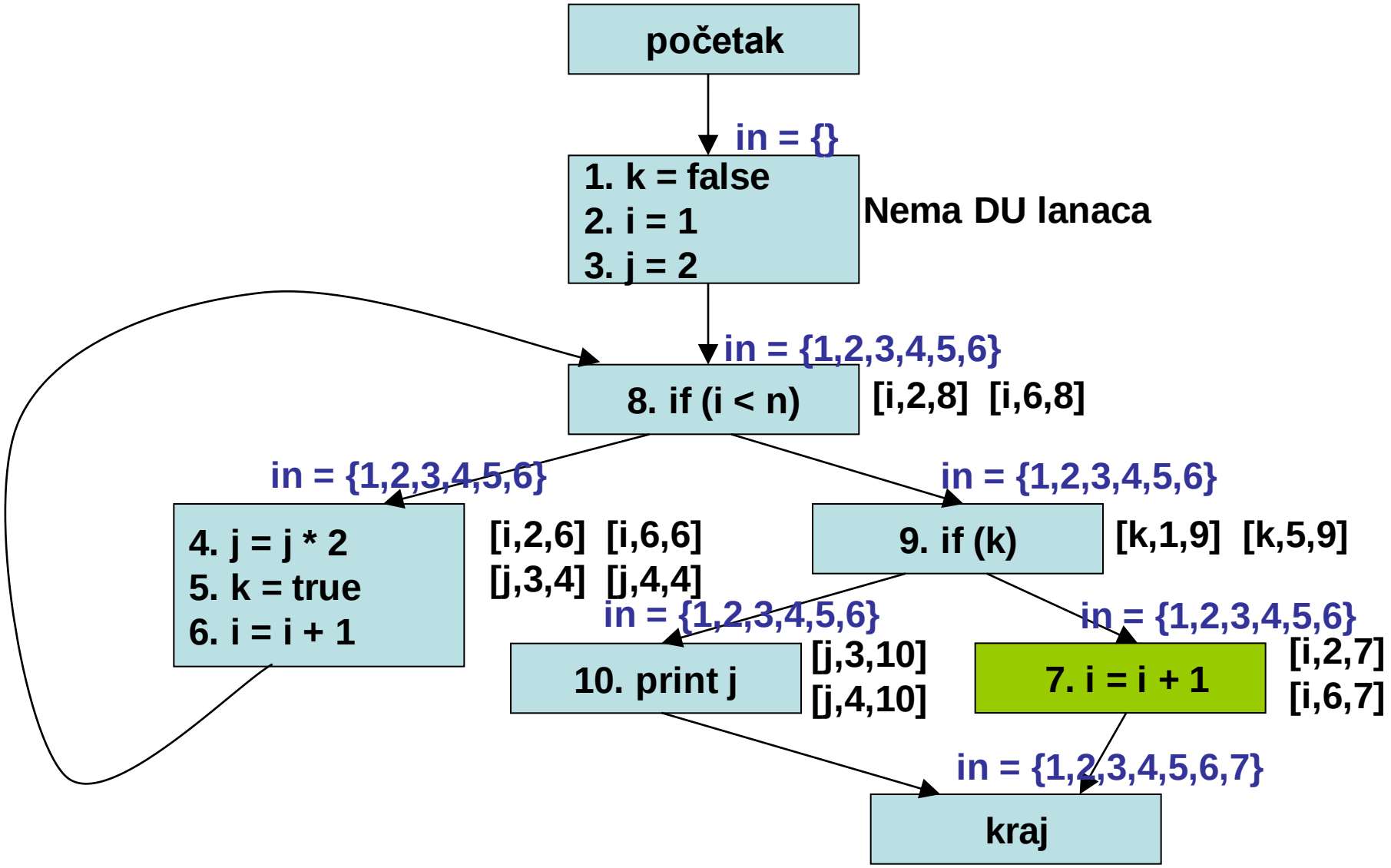
Primer


početak
in = {}
1. k = false
2. i = 1
3. j = 2
Nema DU lanaca
in = {1,2,3,4,5,6}
8. if (i < n)
[i,2,8] [i,6,8]
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
[i,2,6] [i,6,6]
[j,3,4] [j,4,4]
in = {1,2,3,4,5,6}
9. if (k)
[k,1,9] [k,5,9]
in = {1,2,3,4,5,6}
10. print j
in = {1,2,3,4,5,6}
7. i = i + 1
in = {1,2,3,4,5,6,7}
kraj

Primer


početak
in = {}
1. k = false
2. i = 1
3. j = 2
Nema DU lanaca
in = {1,2,3,4,5,6}
8. if (i < n)
[i,2,8] [i,6,8]
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
[i,2,6] [i,6,6]
[j,3,4] [j,4,4]
in = {1,2,3,4,5,6}
9. if (k)
[k,1,9] [k,5,9]
in = {1,2,3,4,5,6}
10. print j
[j,3,10]
[j,4,10]
in = {1,2,3,4,5,6}
7. i = i + 1
in = {1,2,3,4,5,6,7}
kraj

Primer


početak
in = {}
1. k = false
2. i = 1
3. j = 2
Nema DU lanaca
in = {1,2,3,4,5,6}
8. if (i < n)
[i,2,8] [i,6,8]
in = {1,2,3,4,5,6}
4. j = j * 2
5. k = true
6. i = i + 1
[i,2,6] [i,6,6]
[j,3,4] [j,4,4]
in = {1,2,3,4,5,6}
9. if (k)
[k,1,9] [k,5,9]
in = {1,2,3,4,5,6}
10. print j
[j,3,10]
[j,4,10]
in = {1,2,3,4,5,6}
7. i = i + 1
[i,2,7]
[i,6,7]
in = {1,2,3,4,5,6,7}
kraj

Ostale primene globalne analize toka podataka (kod kompajlera)

- Žive promenljive (Live variables)
  - Određivanje da li se promenljiva koristi na putanju od zadate tačke do izlaza
- Raspoloživi izrazi (Available expressions)
  - Određivanje koji izrazi su već izračunati u kojoj tački
- Uposleni izrazi (Very busy expressions)
  - Određivanje da li se izraz izračunava na svim putanjama od zadate tačke do izlaza

Žive promenljive

- Promenljiva x je *živa* u tački p programa ako postoji putanja od p do izlaza po kojoj se vrednost x upotrebljava pre dodele nove vrednosti x-u.
- U suprotnom, promenljiva je *mrtva* u posmatranoj tački.
- Koristi se u :
  - Dodeli registara promenljivima
  - Eliminaciji “mrtvog koda”

Raspoloživi izrazi

- Izraz *x+y* je *raspoloživ u tački p* ako svaka putanja od ulaza do *p* izračunava *x+y* i posle poslednjeg takvog izračunavanja pre dolaska u *p*, nema dodela promenljivama *x* i *y.*
- Koristi se u:
  - Globalnoj eliminaciji zajedničkih podizraza

Uposleni izrazi

- Izraz e je uposlen u tački p ako se, bez obzira po kojoj putanji se ide iza p, e izračunava pre nego što se bilo koji od njegovih operanada redefiniše.
- Koristi se u:
  - Code hoisting
    - Ako je e uposlen u tački p, možemo premestiti njegovo izračunavanje u tačku p.