

LOMLJENA DREVESA

ANDREJA LAPAJNE, HANA
KRANJEC KELBEL

FMF

JANUAR, 2022



ZAČETKI

➤ Ustvaril Sleator, Terjan leta 1985

➤ IDEJA :

Vsakič, ko dostopamo do elementa, ga „zlomimo“ v korenu

Vsakič, ko dostopamo do elementa, ga premaknemo naprej



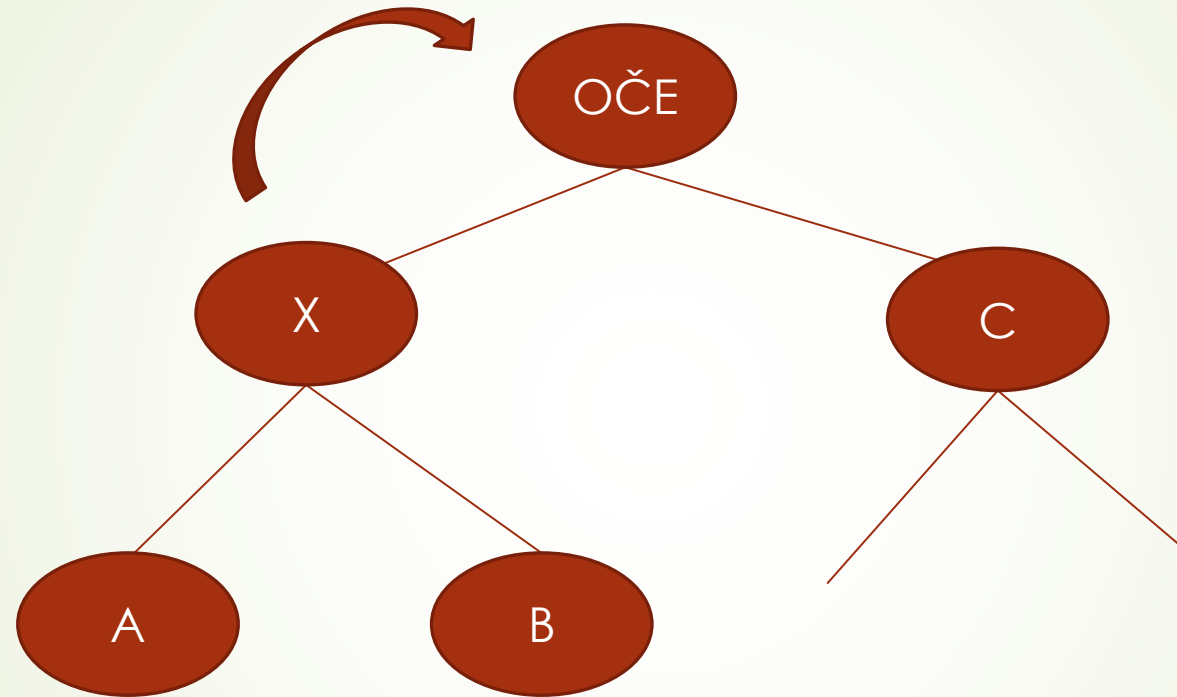
Kaj so lomljena drevesa?

- Se lomijo ob vsakem posegu v drevo
- Lomimo z rotacijami

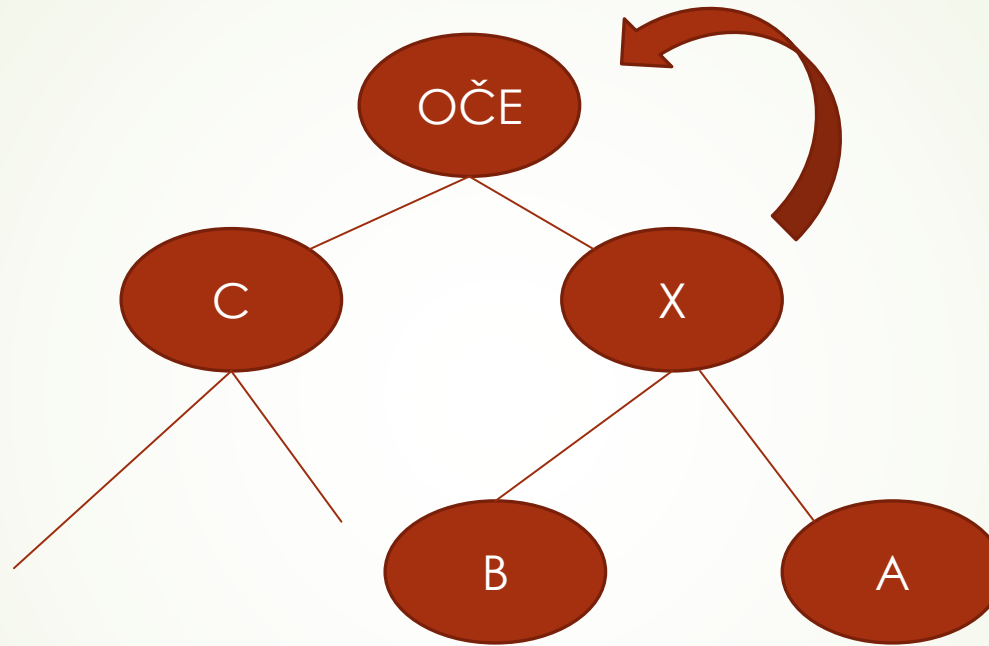
Rotacije

- Leve in desne
- Enojne in dvojne
- Ohranjanje lastnosti iskalnosti drevesa
- 6 tipov rotacij

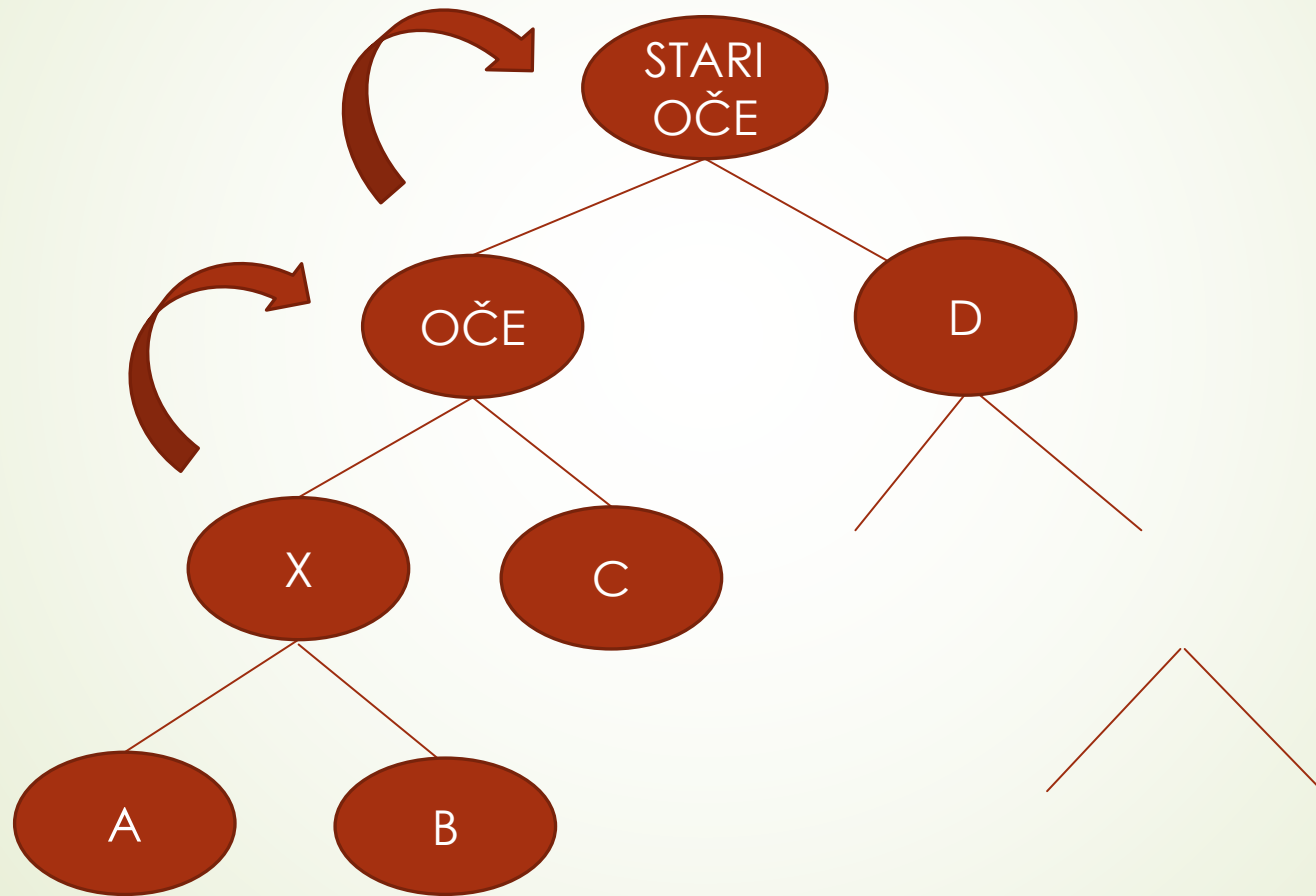
Desna rotacija



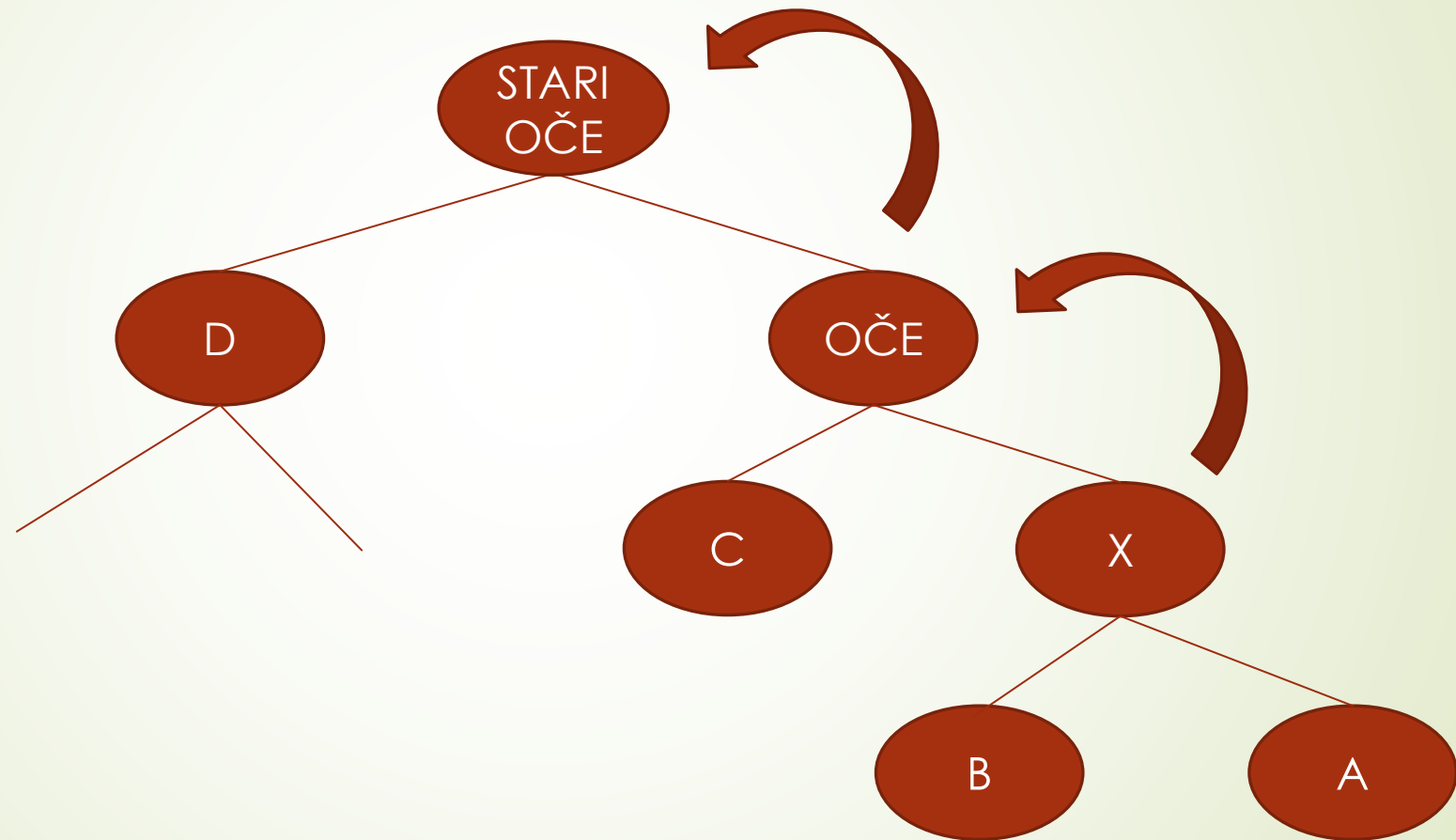
Leva rotacija

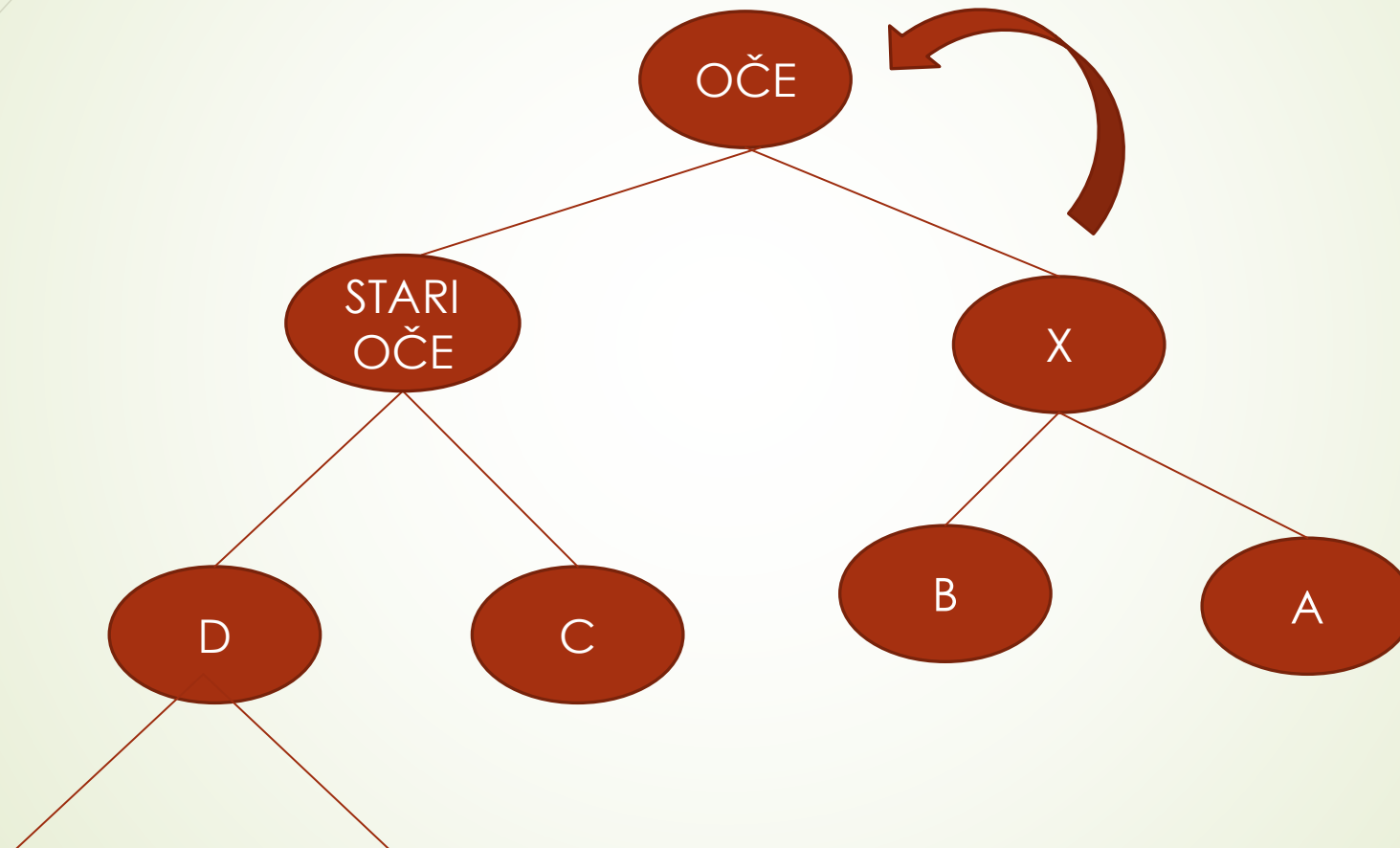


Desno – desna rotacija

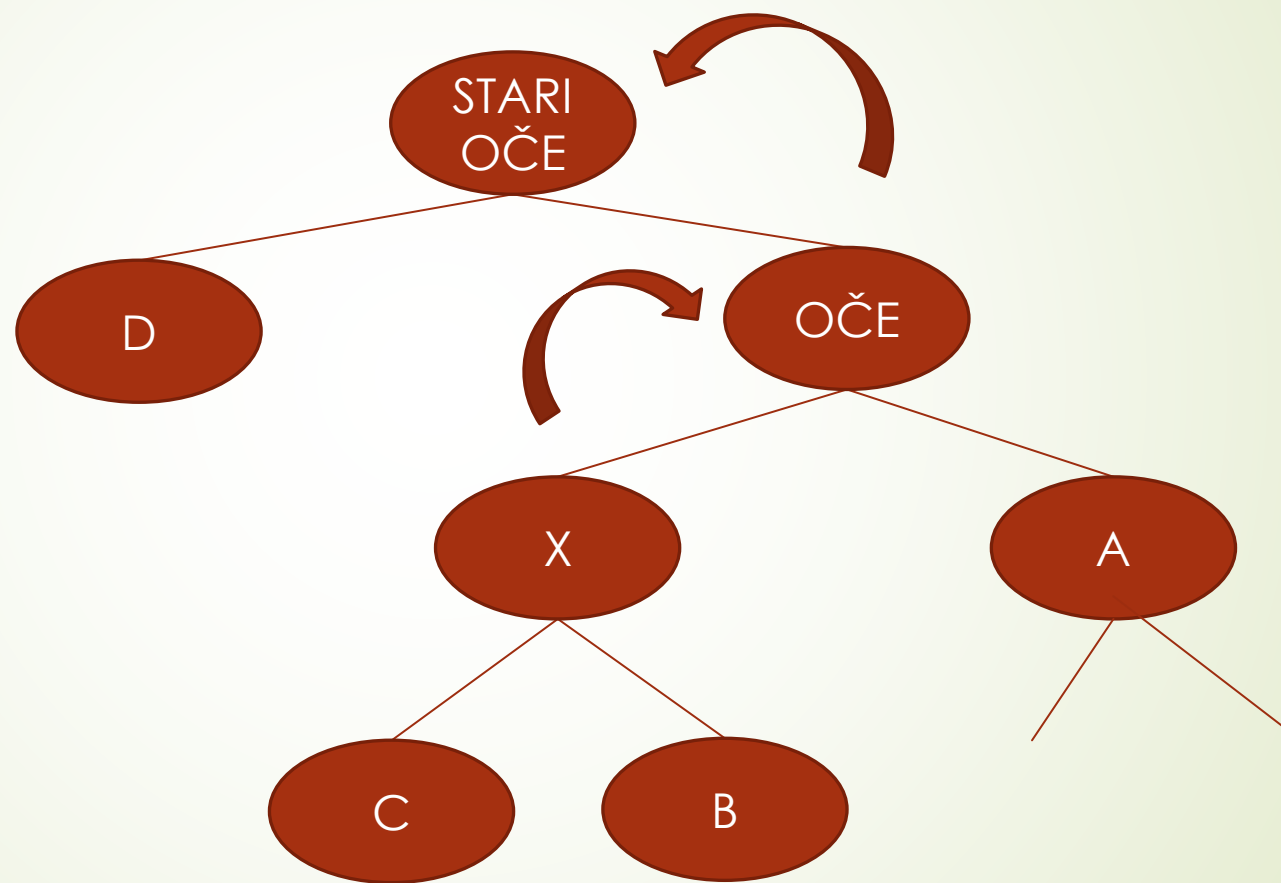


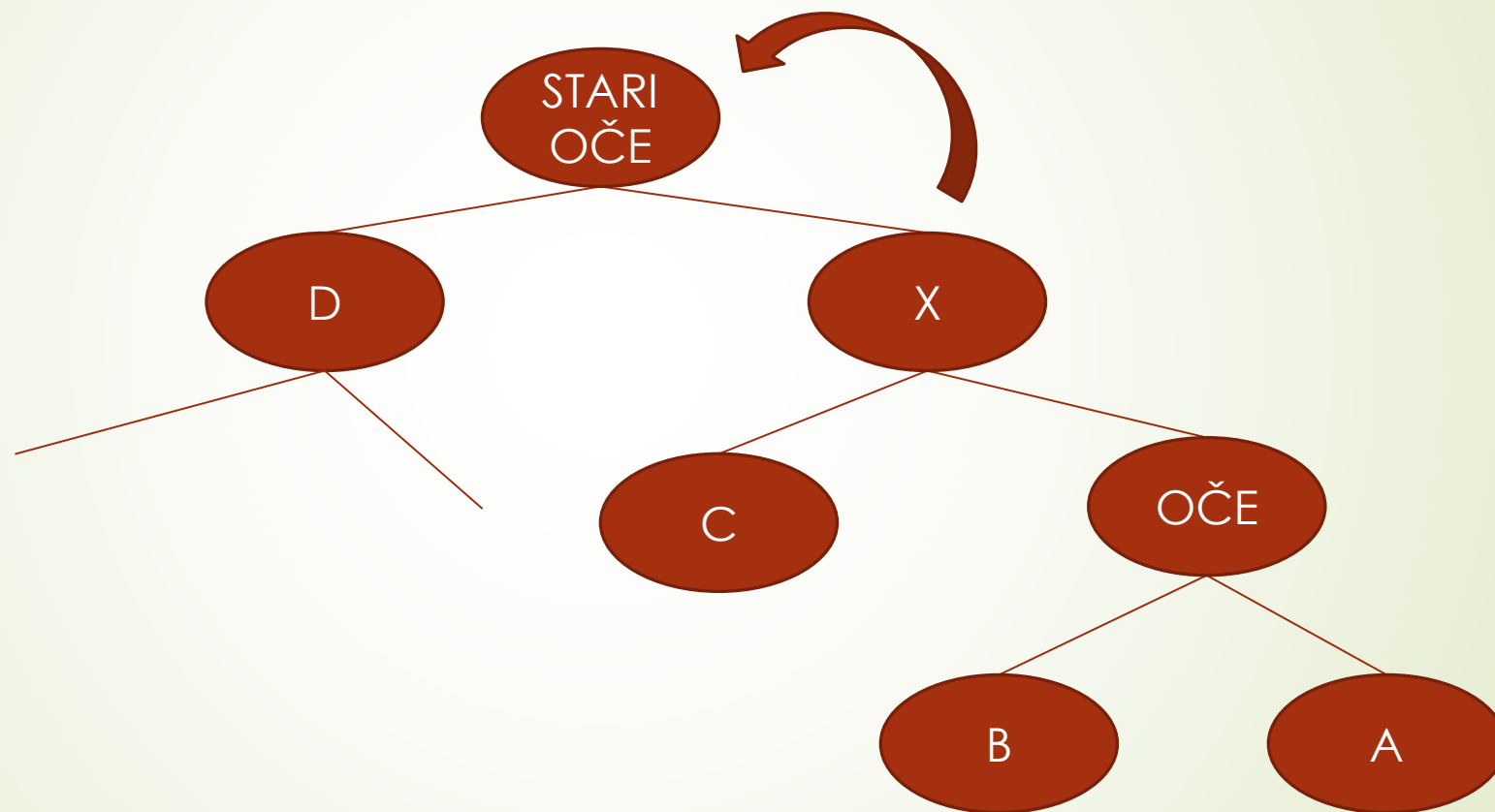
Levo – leva rotacija



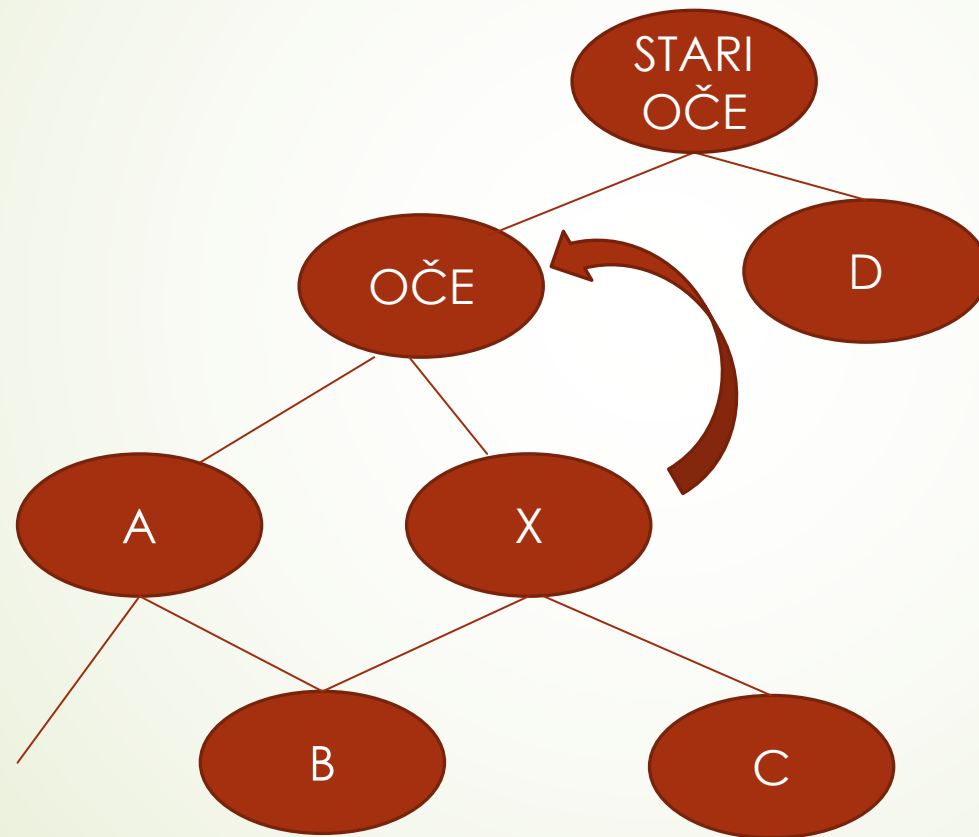


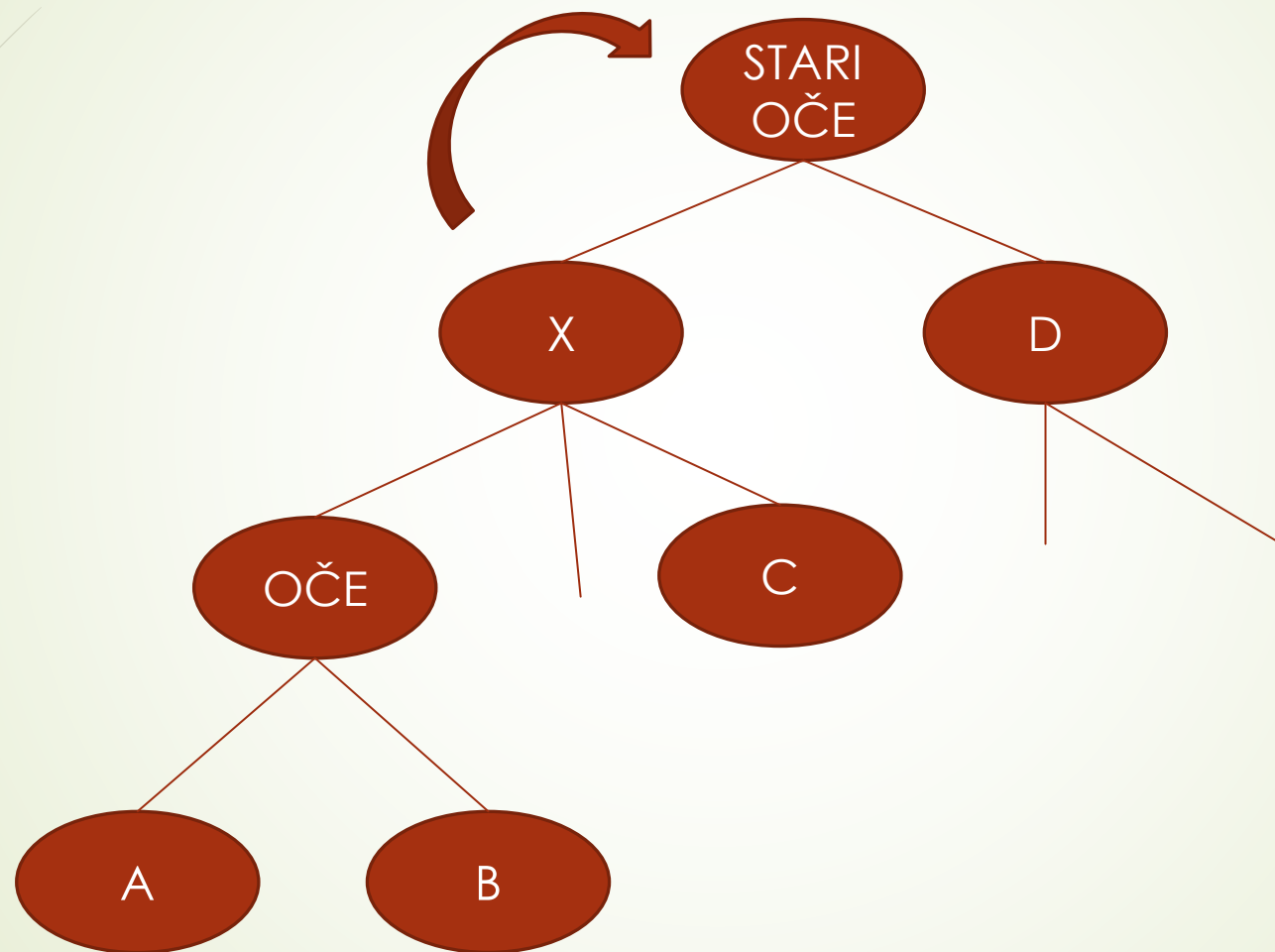
Desno – leva rotacija





Levo – desna rotacija







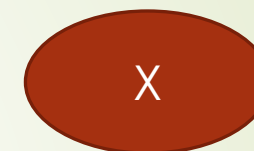
Delitev rotacij

- Dva robna primera
 - Dva splošna primera
- 



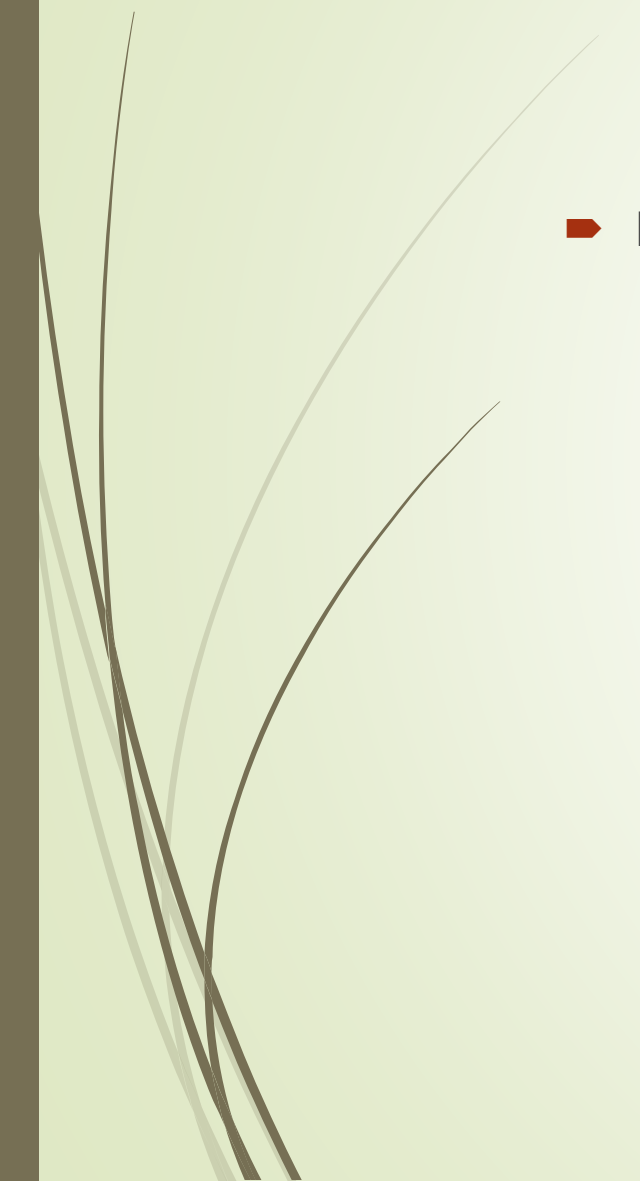
1. robni primer

- Koren drevesa je ustrezno vozlišče

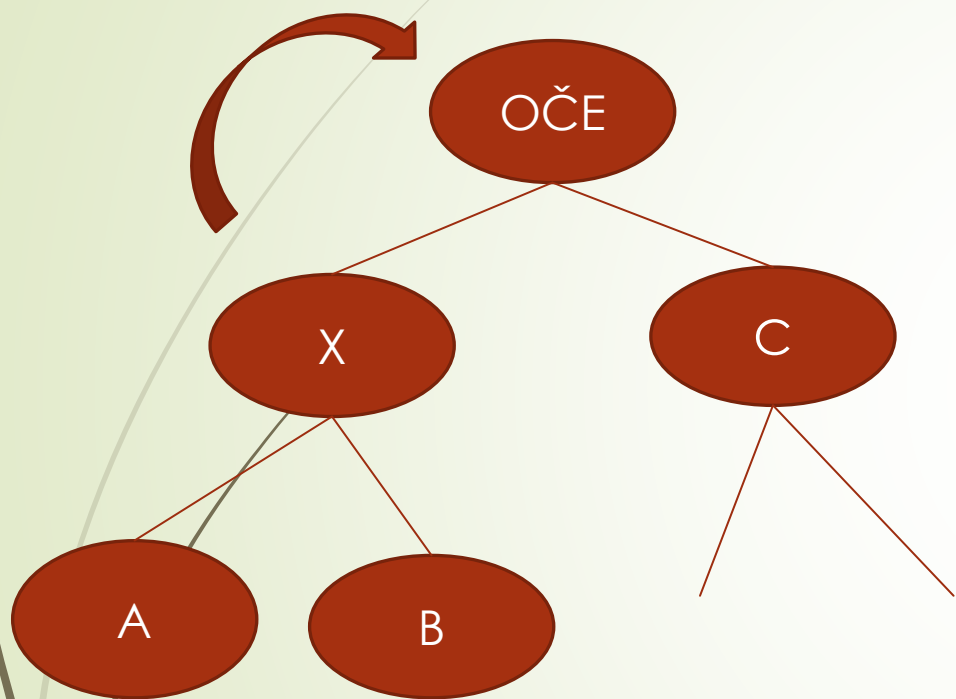




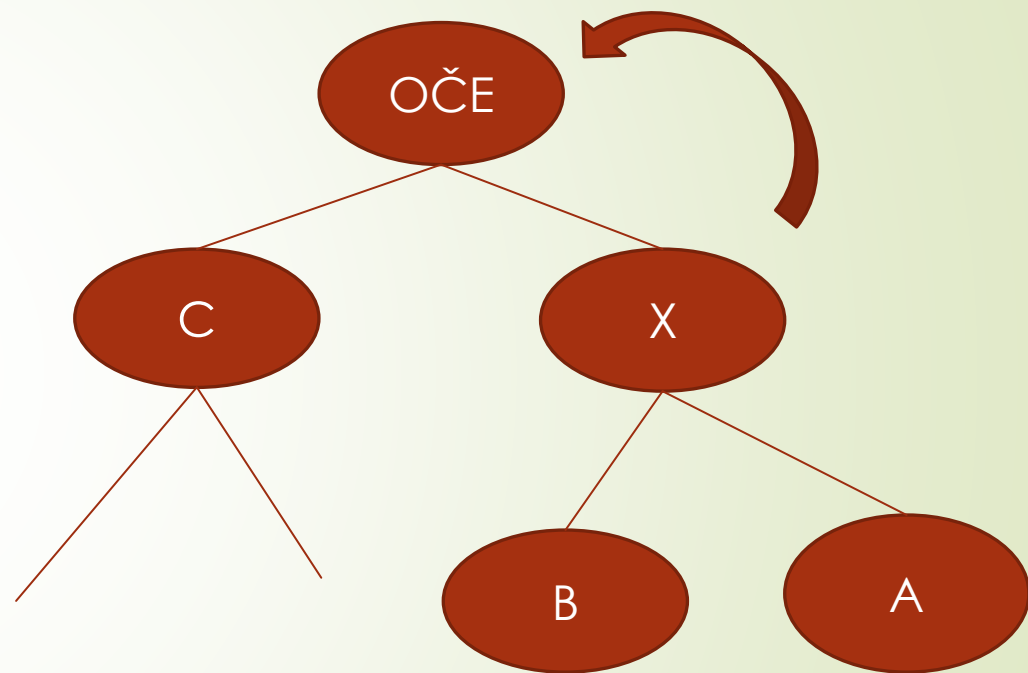
2. robni primer

- Koren drevesa je oče obravnavanega vozlišča
- 

Desna rotacija

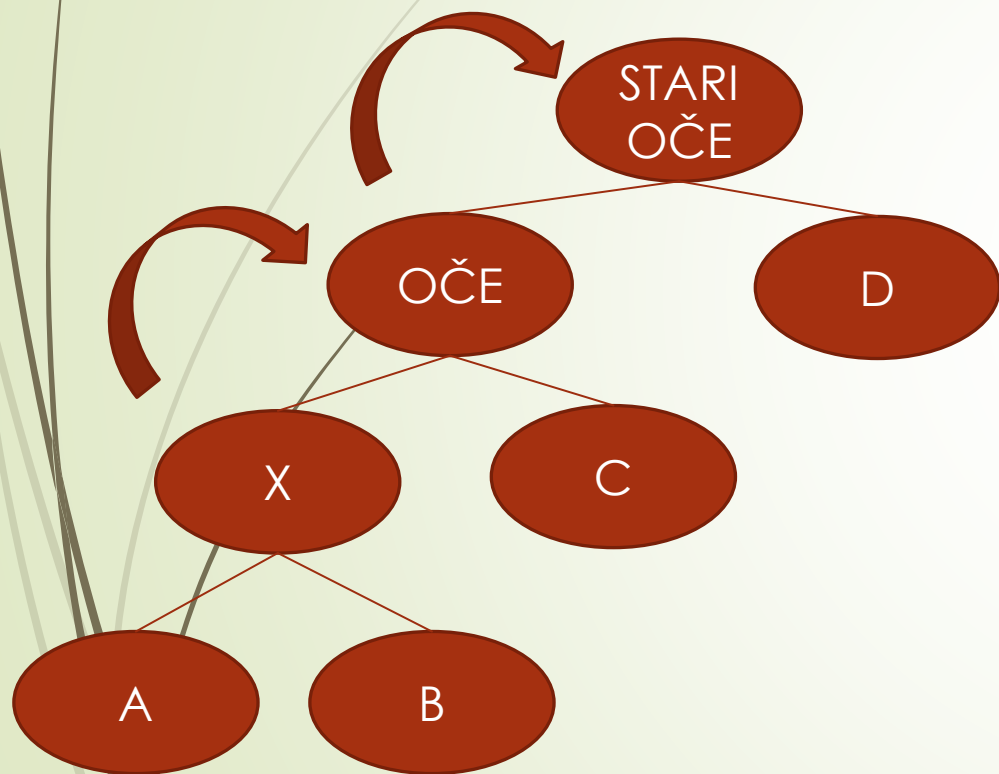


Leva rotacija

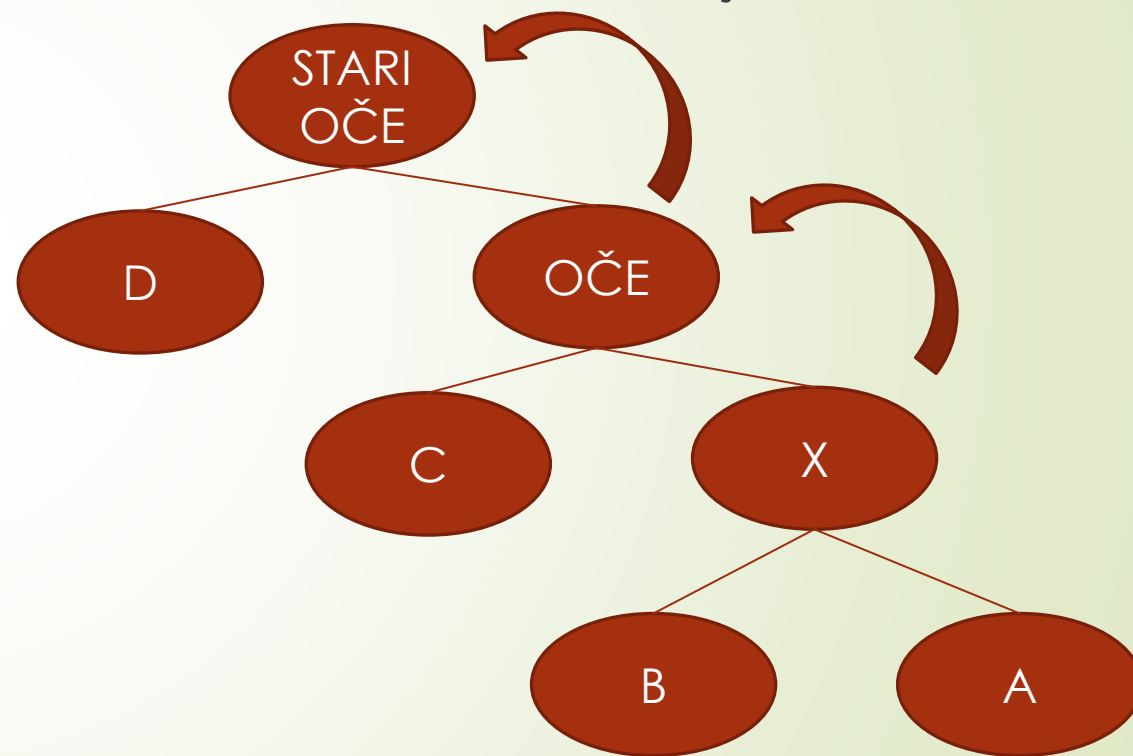


1. Splošen primer

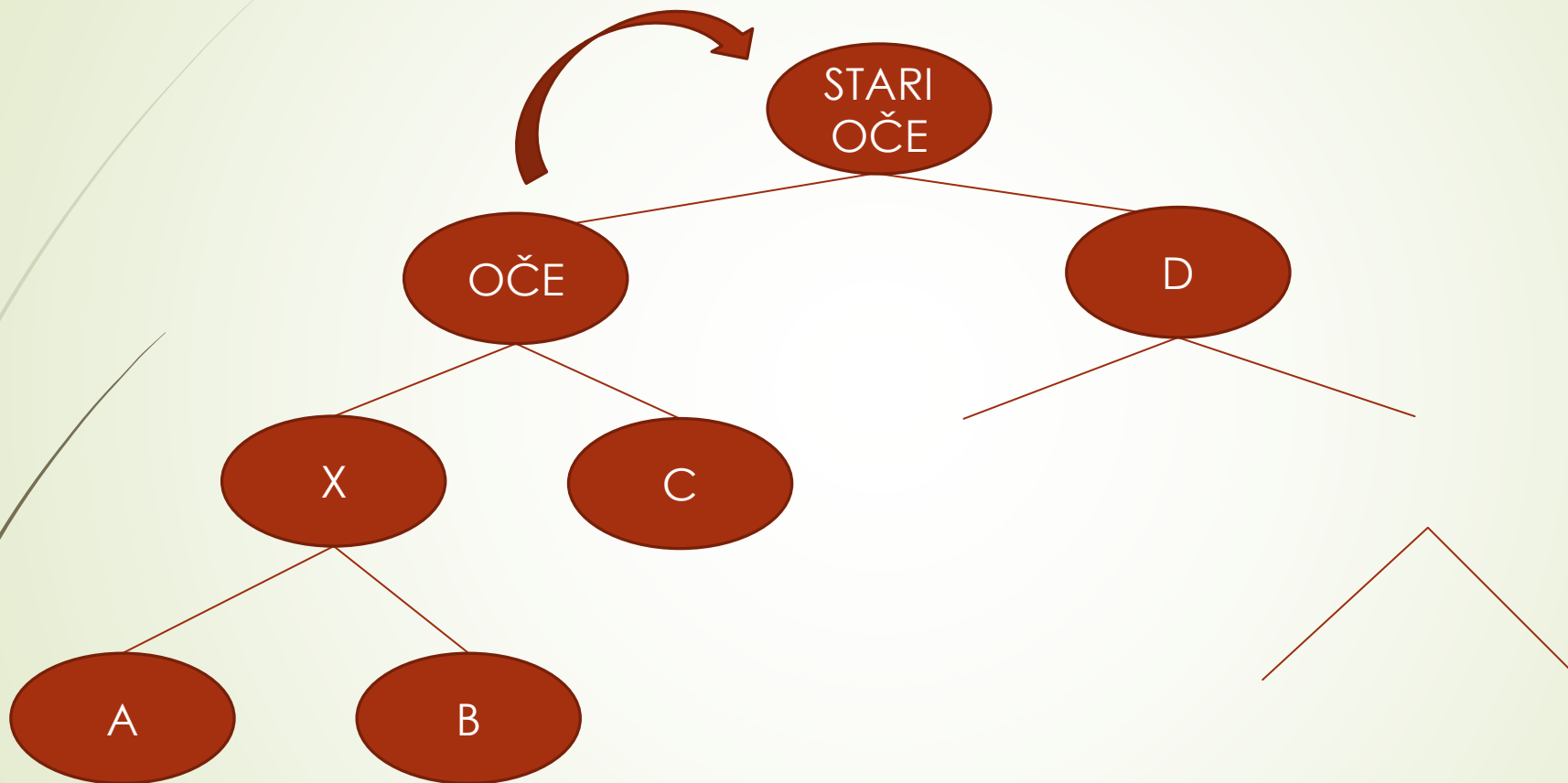
Desna-desna rotacija



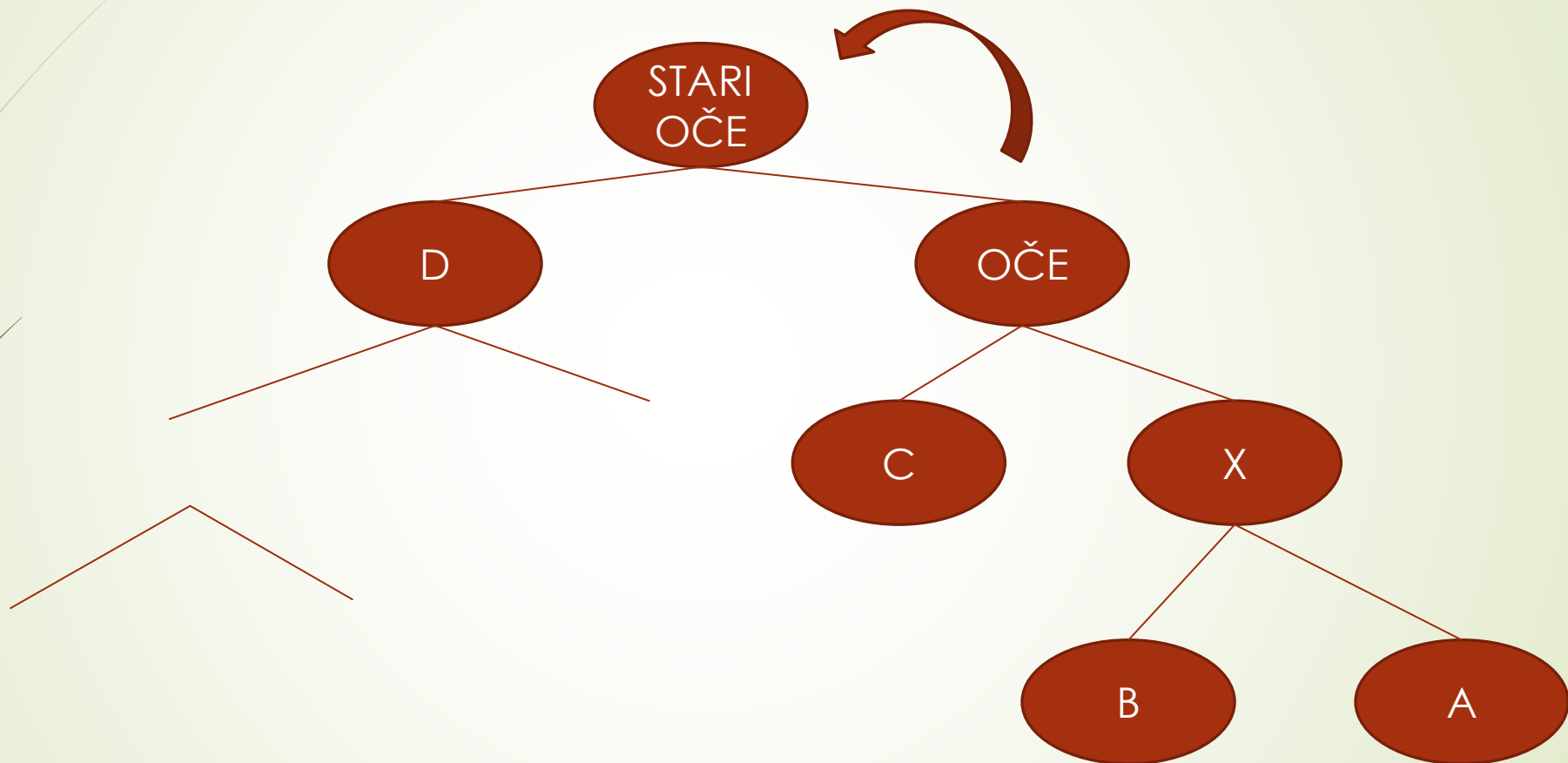
Leva-leva rotacija



Desno – desna rotacija

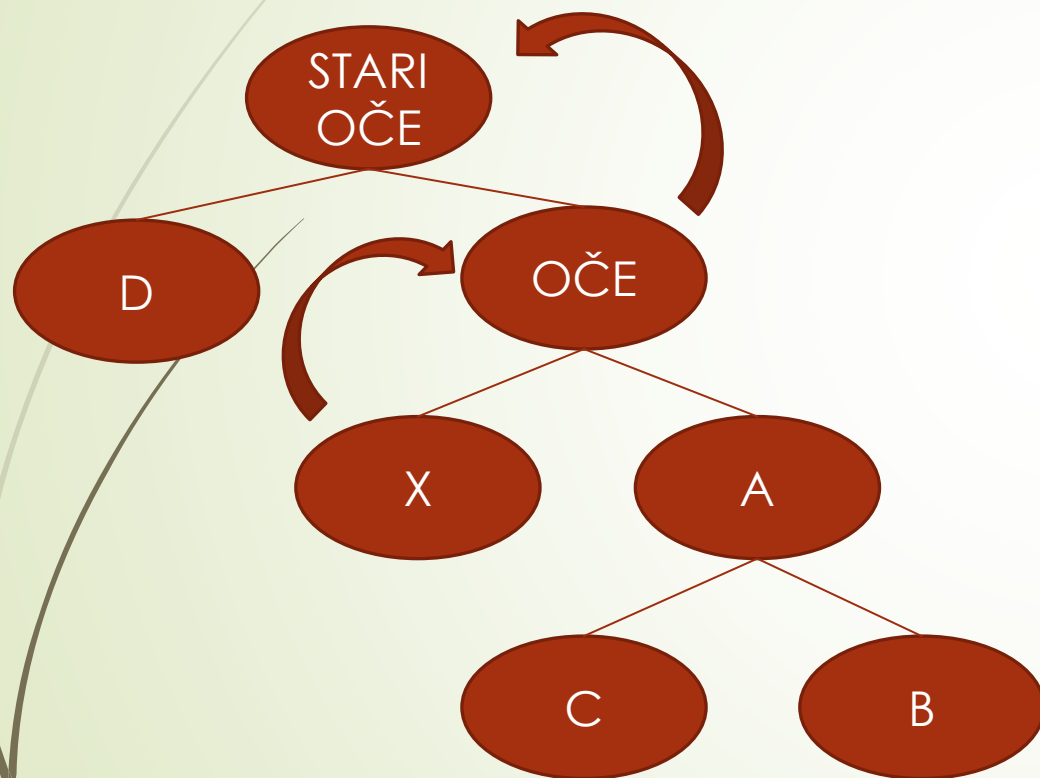


levo – leva rotacija

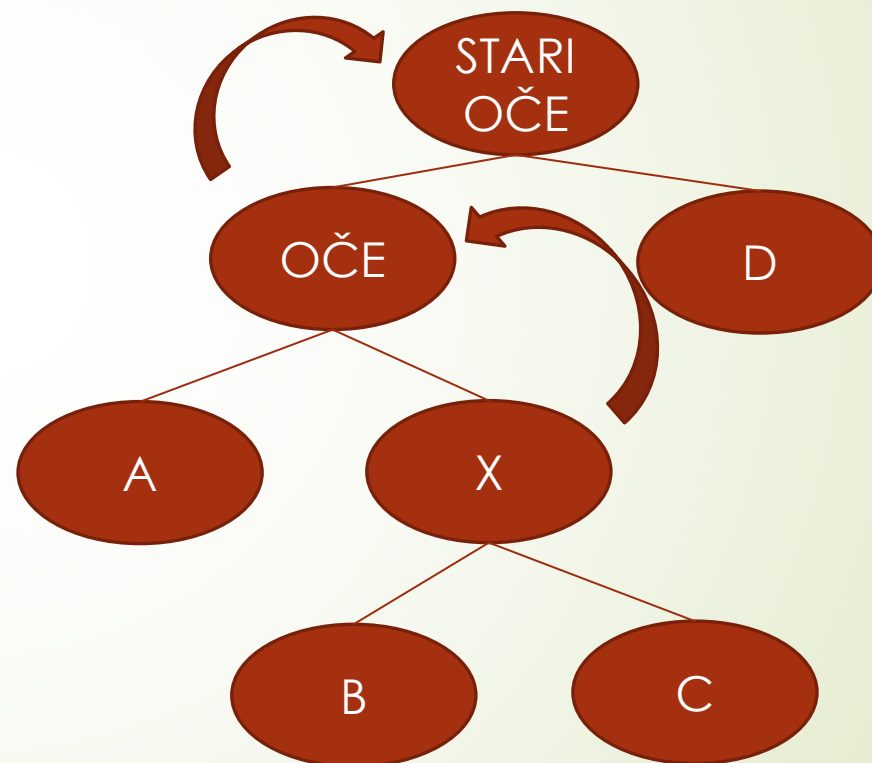


2. Splošni primer

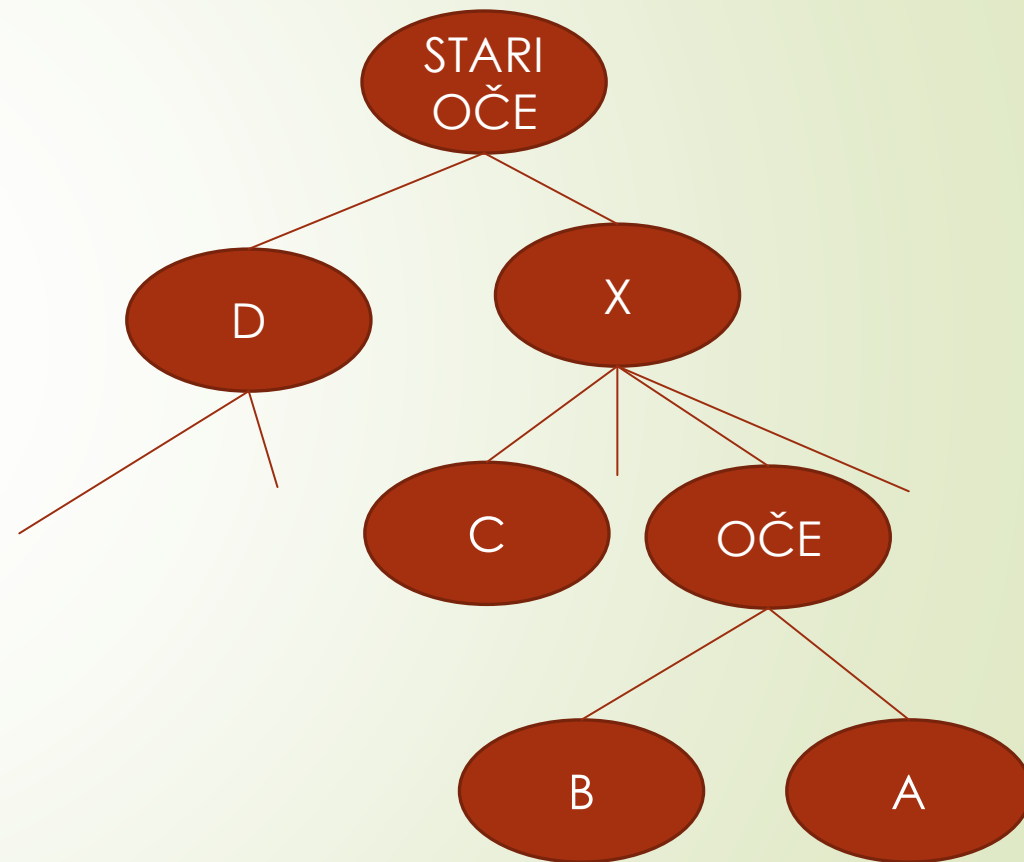
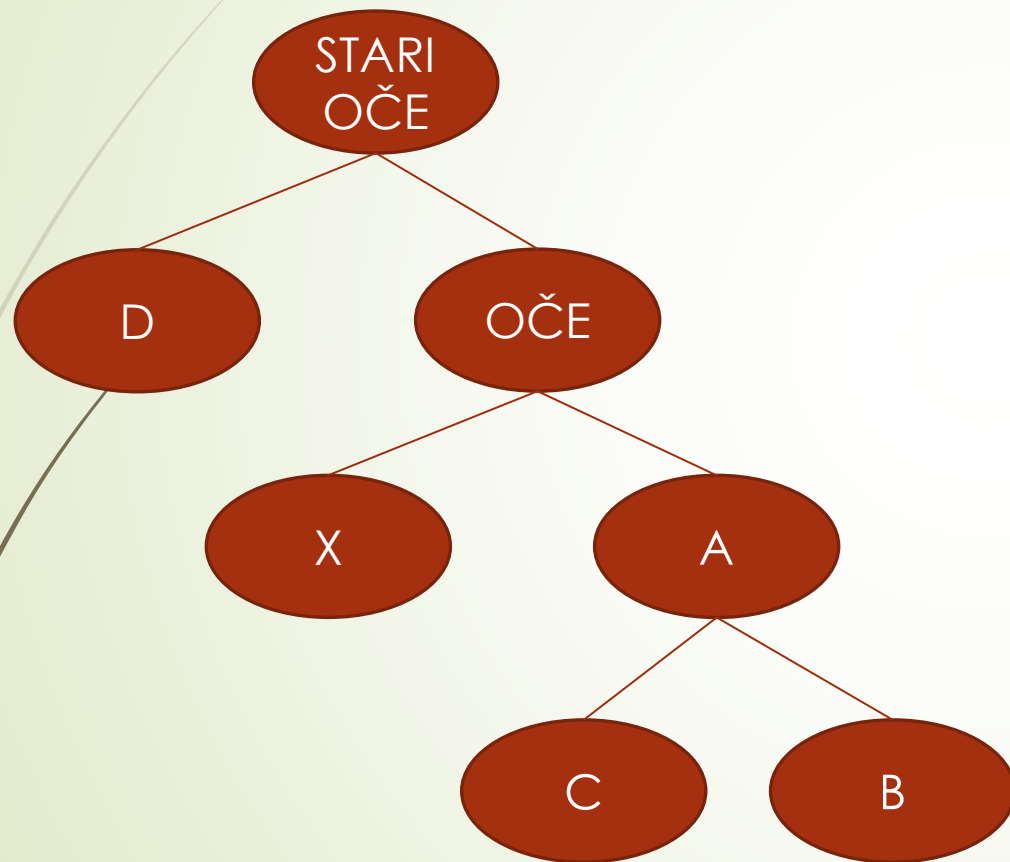
Desno – leva rotacija



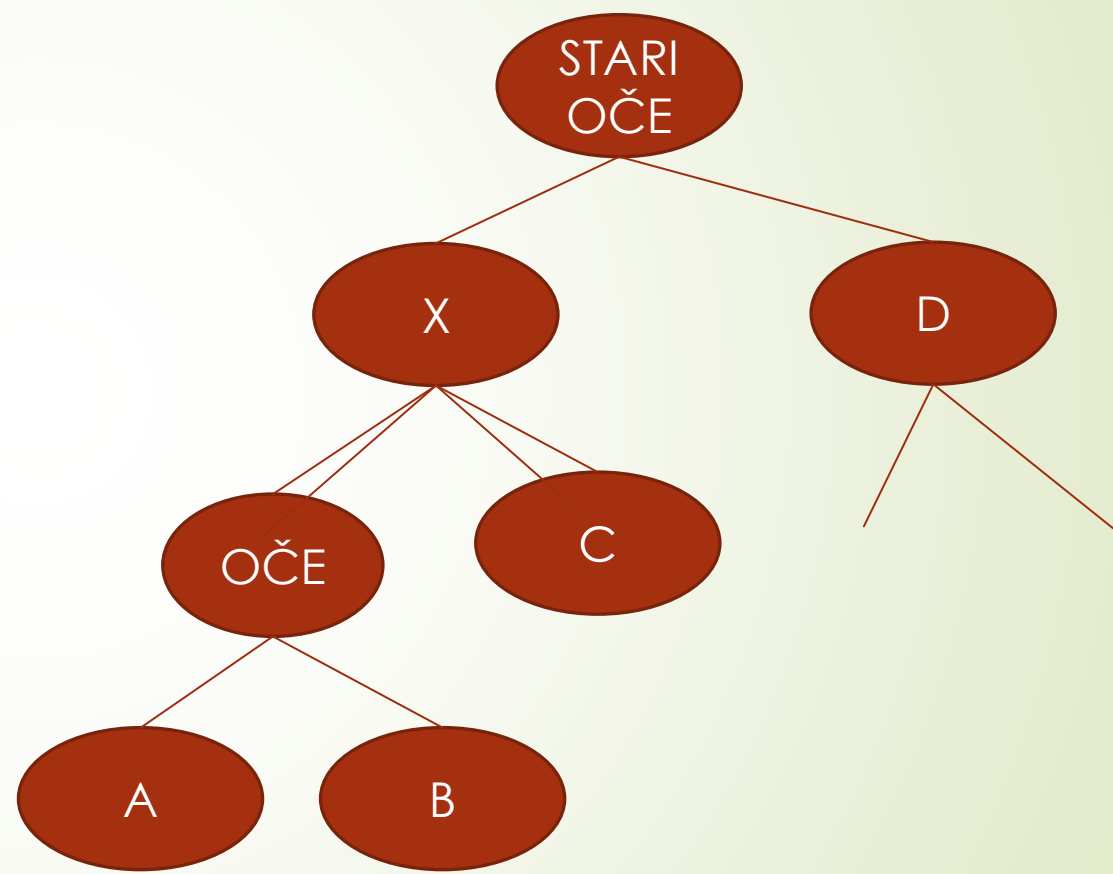
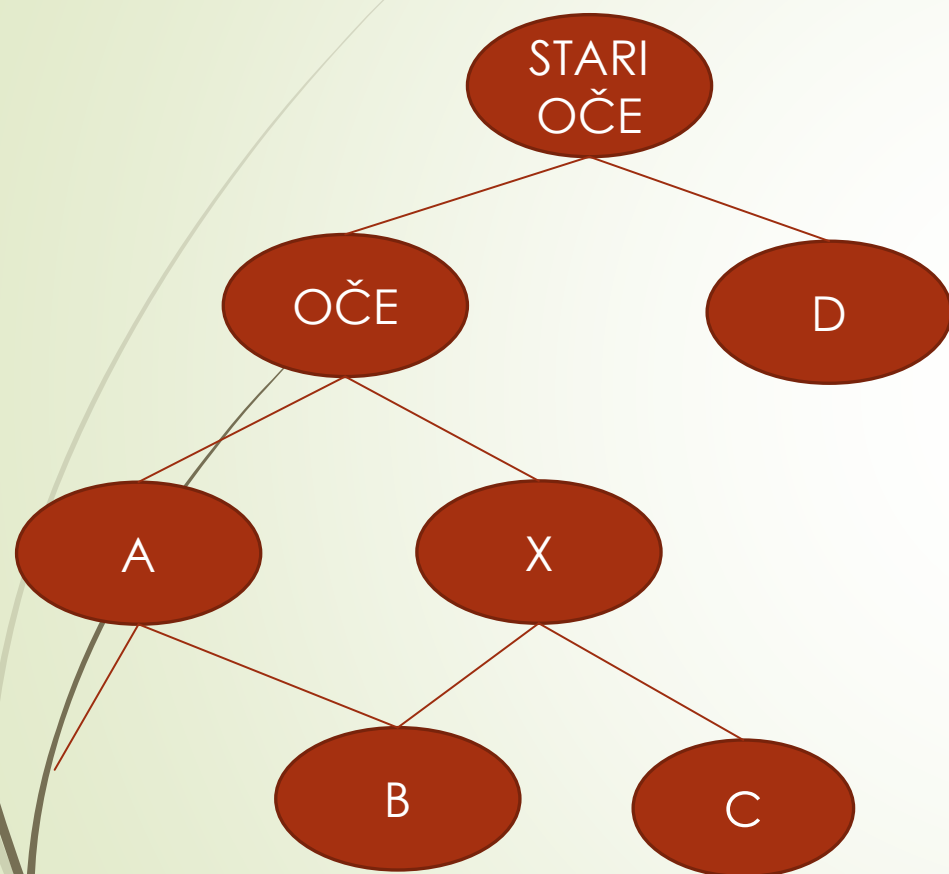
Levo – desna rotacija



Desno – leva rotacija



Levo – desna rotacija



implementacija

```
Lomljenje_drevesa(x) # obravnavan element želimo premakniti v koren drevesa
    while x.oče NOT NULL #izvajamo zanko dokler x ne postane koren drevesa oz.
dokler ima očeta
        if x.oče.levi == NULL #če je stari oče X element NULL oz. element X je
koren levega ali desnega poddrevesa
            if x == x.oče.oče.levi # če je X element levo poddrevo očeta
                // desna rotacija
                desna-rotacija(x.oče)
            else
                // leva rotacija # če je pa X element desno poddrevo naredimo
levo rotacijo
                leva-rotacija(x.oče)
            elseif x==x.oče.levi and x.oče == x.oče.oče.levi #če je X očetovo levo
poddrevo in prav tako levo poddrevo starega očeta
                // desno-desna rotacija
                desna-rotacija(x.oče)
                desna-rotacija(x.oče)
            elseif x==x.oče.desni and x.oče == x.oče.oče.desni #če je X očetovo in
staro-očetovo desno poddrevo
                // levo-levo rotacijo
                leva-rotacija(x.oče)
                leva-rotacija(x.oče)
            elseif x==x.oče.desni and x.oče == x.oče.oče.levi #če je X očetovo levo
desno poddrevo in levo poddrevo starega očeta
                // desno-leva rotacija
                leva-rotacija(x.oče)
                desna-rotacija(x.oče)
        else
            // levo-desna rotacija # X je očetovo levo poddrevo in desno
poddrevo starega očeta
            desna-rotacija(x.oče)
            leva-rotacija(x.oče)
    ...
```

Vstavljanje elementa v drevo

- Drevo je prazno
- Drevo ni prazno
- Element, ki ga želimo vstaviti je že v drevesu

```
1.korak : Vstavimo element v drevo  
2.korak : prestavimo x v koren drevesa
```

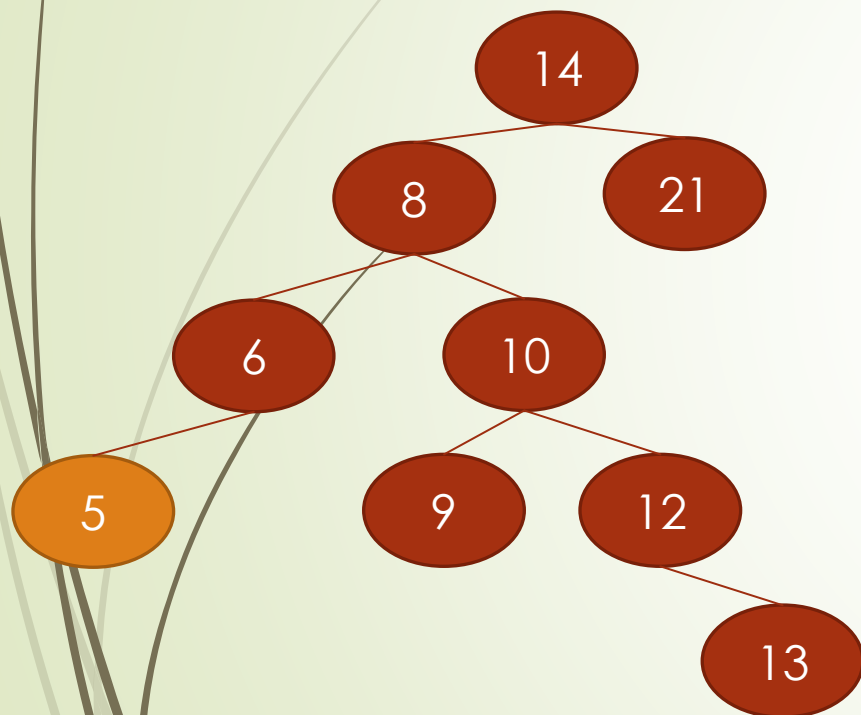
```
vstavi_element(x)  
    vstavi_element_v_drevo(x)  
    Lomljenje_drevesa(x)
```



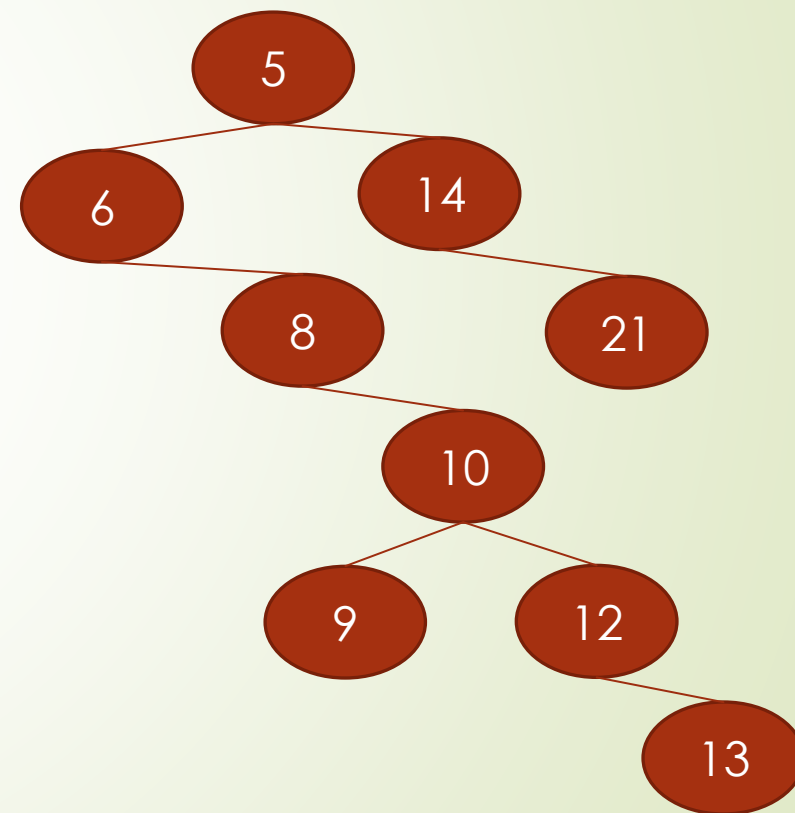
Drevo je prazno
- vstavimo element 5



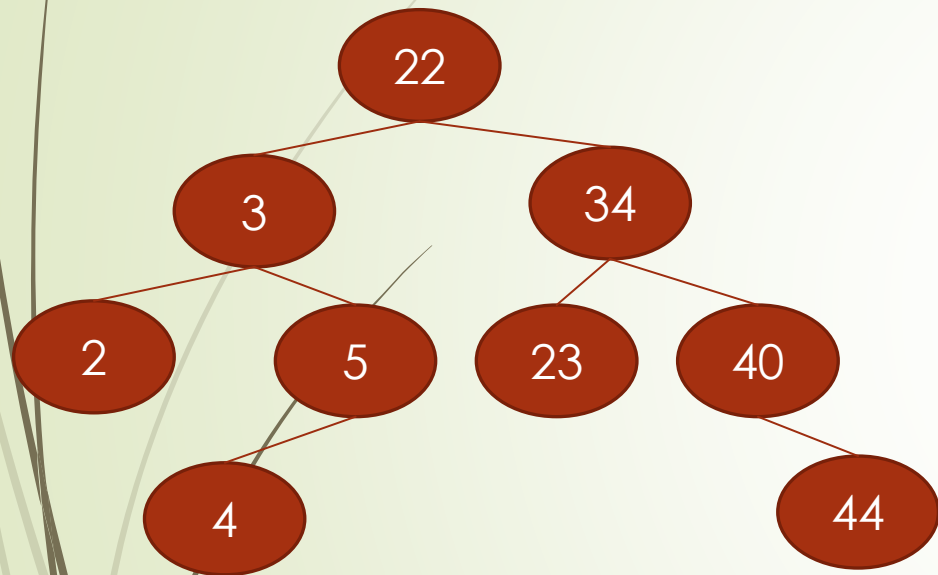
Drevo ni prazno
- vstavimo element 5



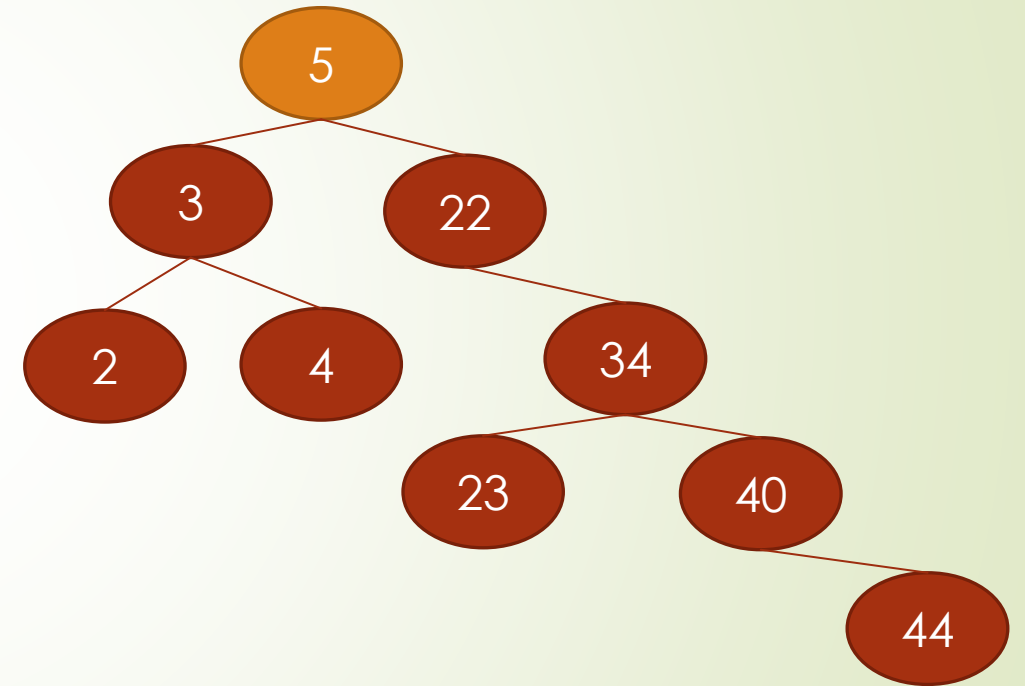
lomljenje →



Element, ki ga želimo vstaviti je v drevesu
- vstavimo 5



lomljenje



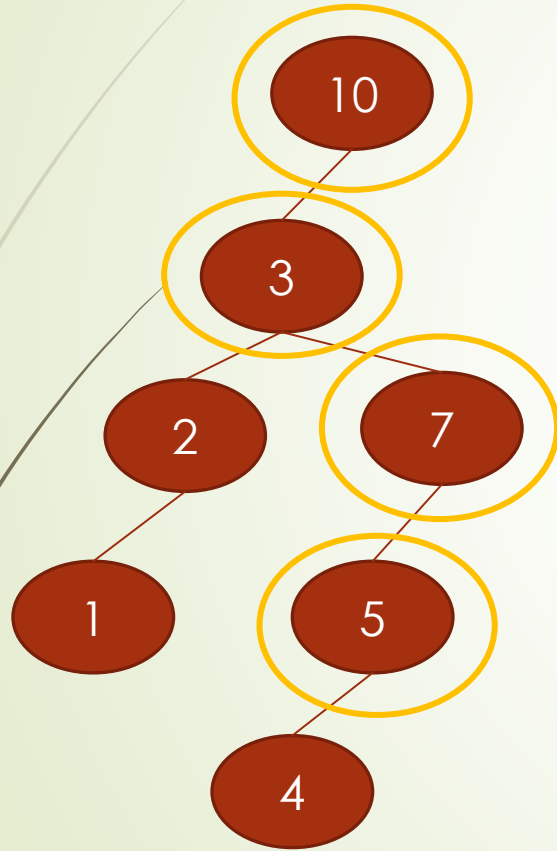
Iskanje elementa v drevesu

- Drevo je prazno
- Našli smo vozlišče z iskanim elementom
- Drevo nima vozlišča z iskanim elementom

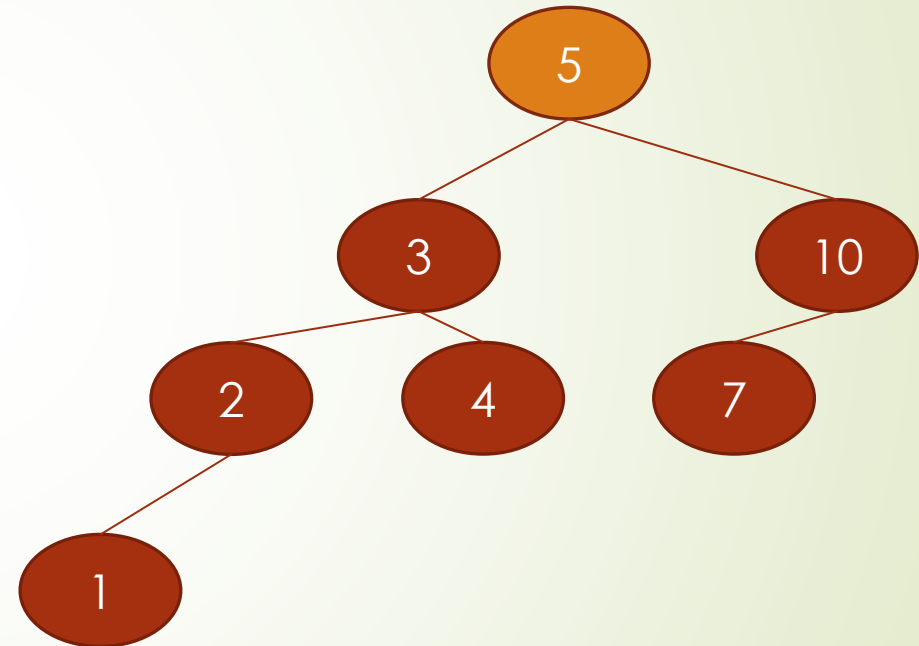
```
1.korak : poiščemo element v drevesu
2. korak : prestavimo x v koren drevesa
poišči_element(ključ)
    x = poišči_element_v_drevesu(ključ)
    if x NOT None # iskan element je v drevesu
        Lomljenje_drevesa(x) # prični z lomljenjem dreveasa
```

Primer iskanje elementa 5

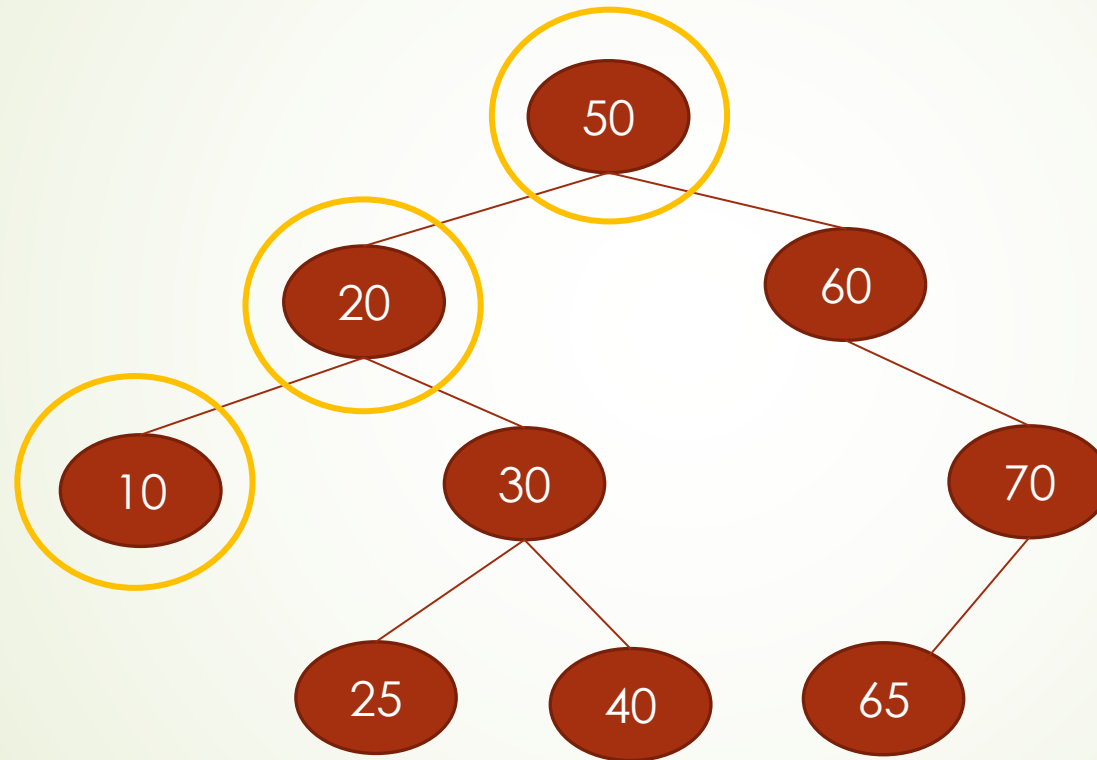
- element je v drevesu



lomljenje



Primer iskanje elementa 5
- drevo nima iskanega elementa/vozlišča



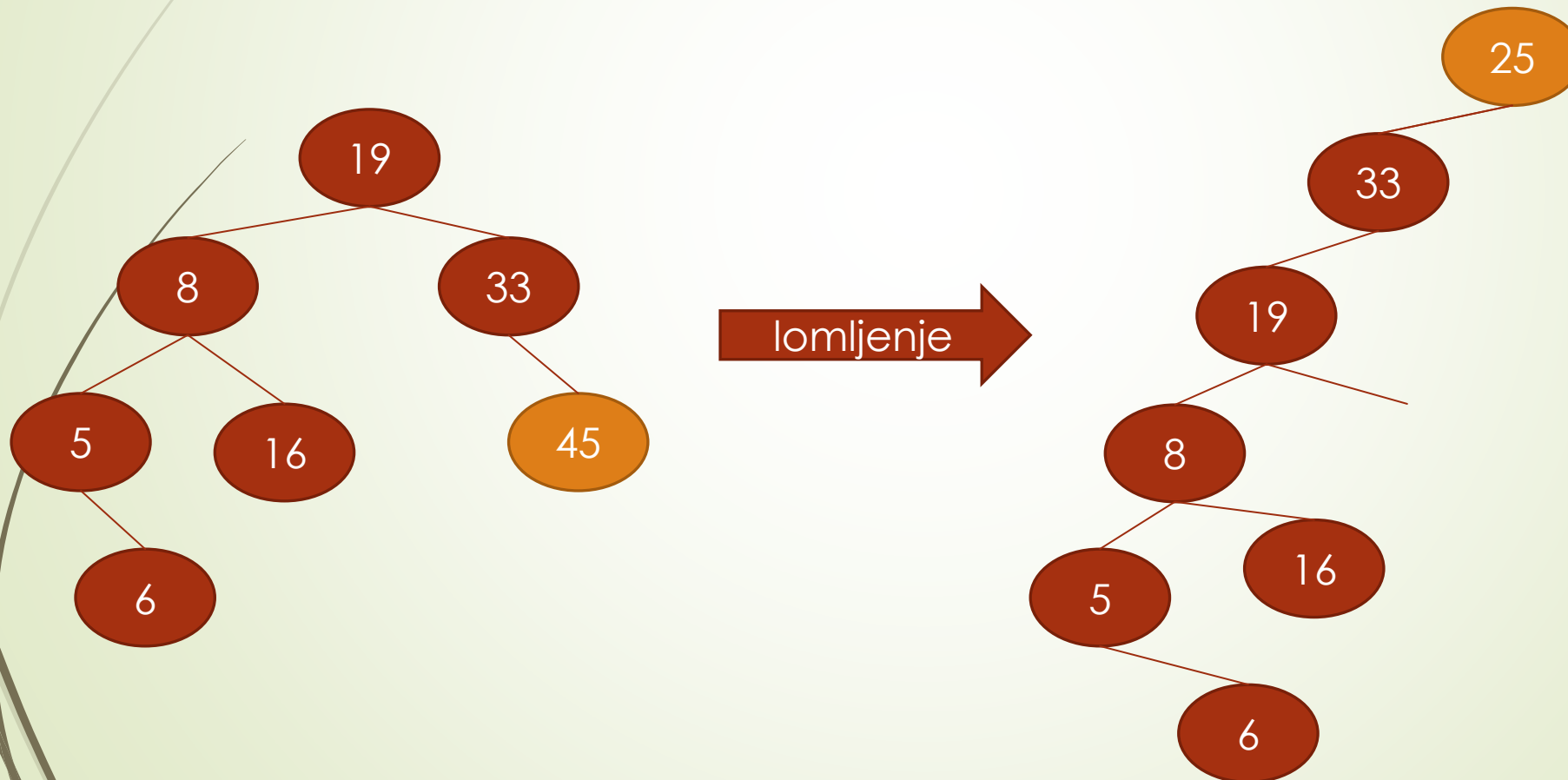
Brisanje elementa v drevesu

- Drevo je prazno
- Drevo ima element, ki ga želimo izbrisati
 - Levo/desno poddrevo je prazno
 - Levo/desno poddrevo ni prazno
- Drevo nima elementa, ki ga želimo izbrisati

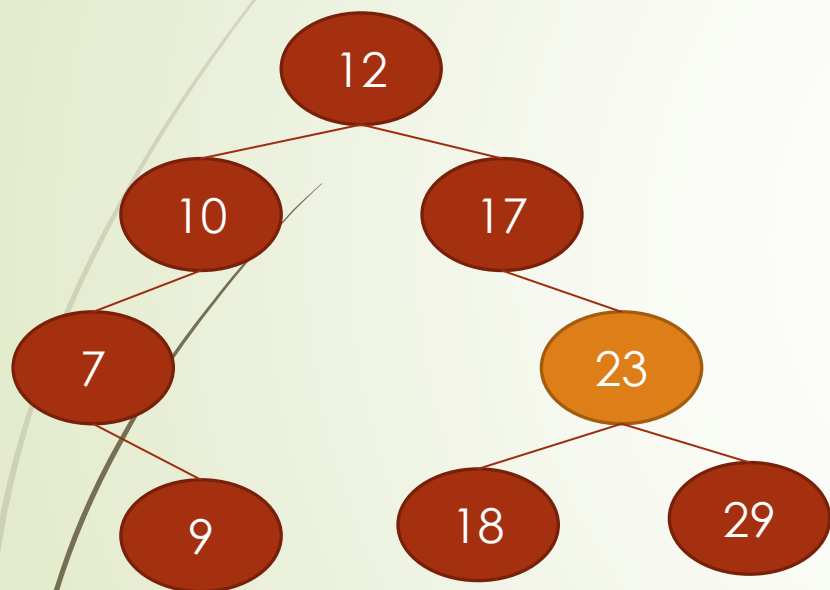
```
1. korak : Zlomimo poddrevo A in B
2. korak : Odstranimo x iz korena drevesa A
3. korak : Združimo A in b
```

```
lomljenje_brisanje(x)
    B, A = lomljenje_drevesa(x)
    if A.levi IS NOT None
        A.levi.oče IS None
    JOIN(A.levi, B)
```

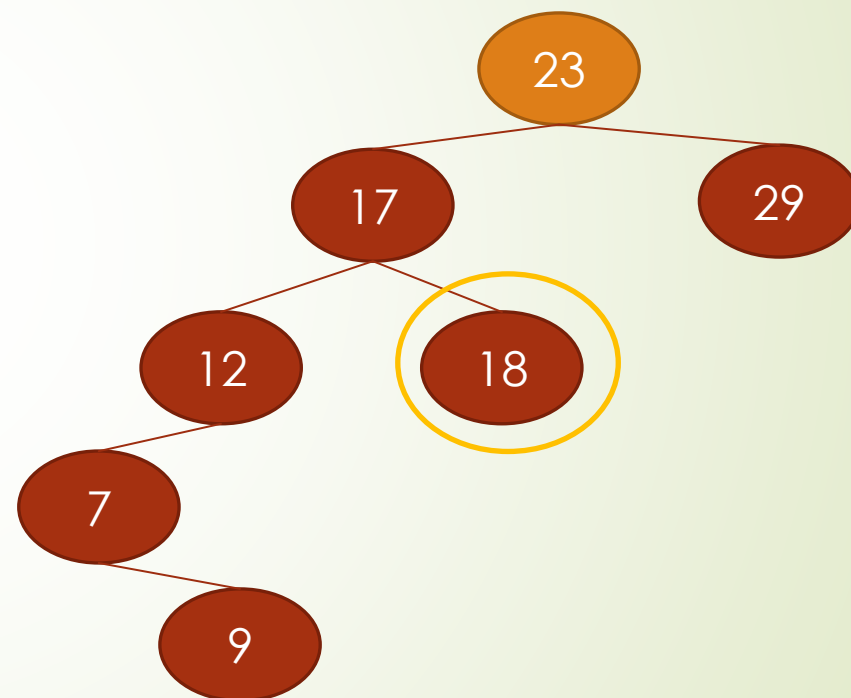
Drevo ima element, ki ga želimo izbrisati
(levo/desno poddrevo je prazno)
- izbrišemo element 25

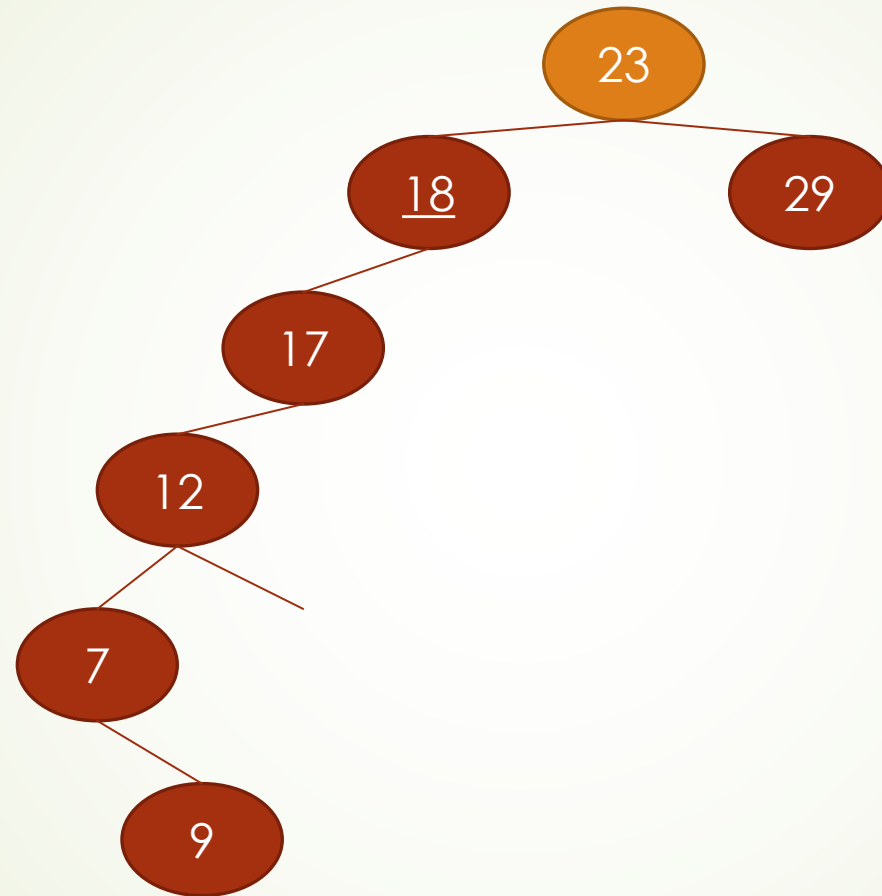


Drevo ima element, ki ga želimo izbrisati
(levo/desno poddrevo ni prazno)
- izbrišemo element 23

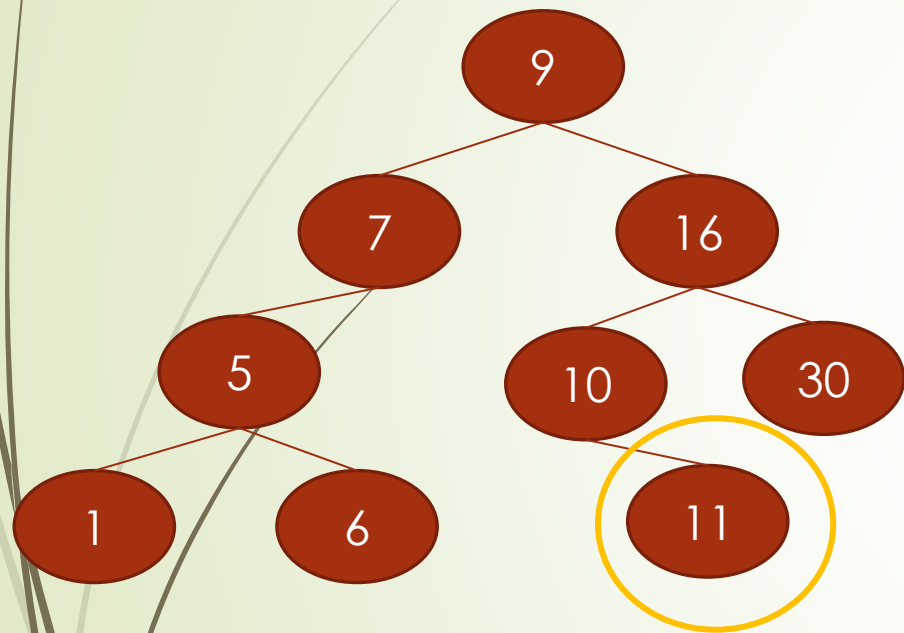


lomljenje

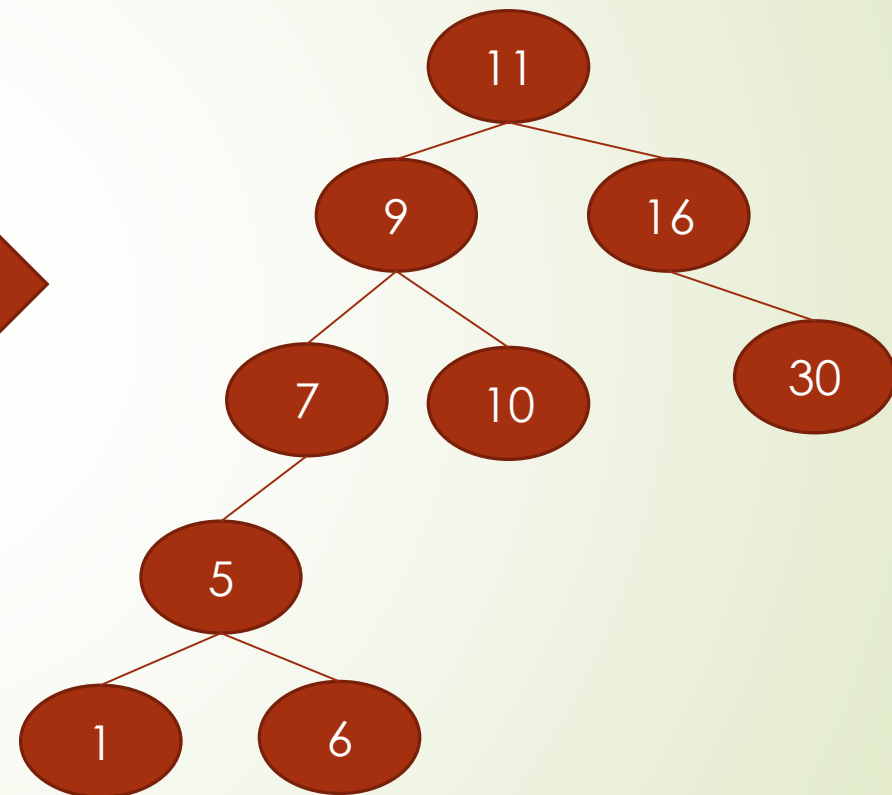




Drevo nima elementa, ki ga želimo izbrisati
- Izbrišimo element 12



lomljenje





Časovna zahtevnost operacij

Lomljenje drevesa in rotacije

- Pričakovana : $O(\log(n))$
- V najslabšem primeru: $O(n)$

Vstavljanje elementa v drevo

- $O(\log(n))$



Časovna zahtevnost

Iskanje elementa v
drevesu

► Pričakovana : $O(\log(n))$

Brisanje elementa v
drevesu

► Pričakovana : $O(\log(n))$

viri

- <https://www.codesdope.com/course/data-structures-splay-trees/> [Dostop:07.01.2022]
- [https://wiki.lokar.fmf.uni-lj.si/psawiki/images/archive/6/64/20090731110137!Lomljena drevesa.pdf](https://wiki.lokar.fmf.uni-lj.si/psawiki/images/archive/6/64/20090731110137!Lomljena_drevesa.pdf)
[Dostop:07.01.2022]
- [https://wiki.lokar.fmf.uni-lj.si/r2wiki/index.php/Lomljena drevesa](https://wiki.lokar.fmf.uni-lj.si/r2wiki/index.php/Lomljena_drevesa) [Dostop:07.01.2022]
- [https://wiki.lokar.fmf.uni-lj.si/r2wiki/images/e/e9/S%26D LomljenoDrevo prosojnice.pdf](https://wiki.lokar.fmf.uni-lj.si/r2wiki/images/e/e9/S%26D_LomljenoDrevo_prosojnice.pdf)
[Dostop:07.01.2022]
- [https://lokar.fmf.uni-lj.si/www/osebno/OpravljeneDiplome/Simon Butalic diploma.pdf](https://lokar.fmf.uni-lj.si/www/osebno/OpravljeneDiplome/Simon_Butalic_diploma.pdf)
[Dostop:07.01.2022]