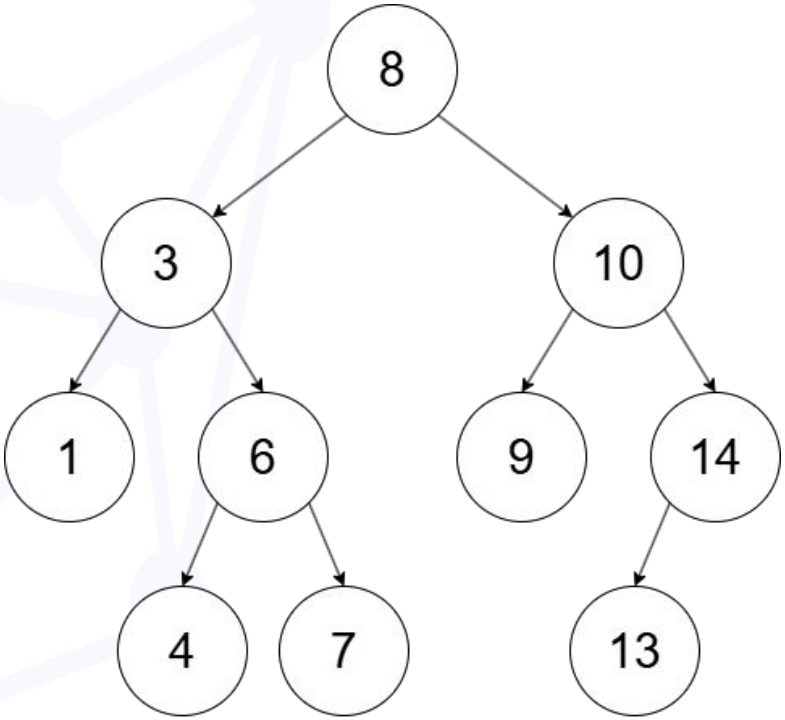
Teorija Algoritama

Balansirana binarna stabla

Binarno stablo traženja - BST

- Za svako podstablo važi:
  - Ključ korena je veći od svih ključeva u levom podstablu;
  - Ključ korena je manji od svih ključeva u desnom podstablu.
- *Inorder* obilazak stabla (left, parent, right) daje niz elemenata sortiran po ključu.
- Osnovne operacije:
  - Dodavanje;
  - Brisanje;
  - Pretraga po ključu.

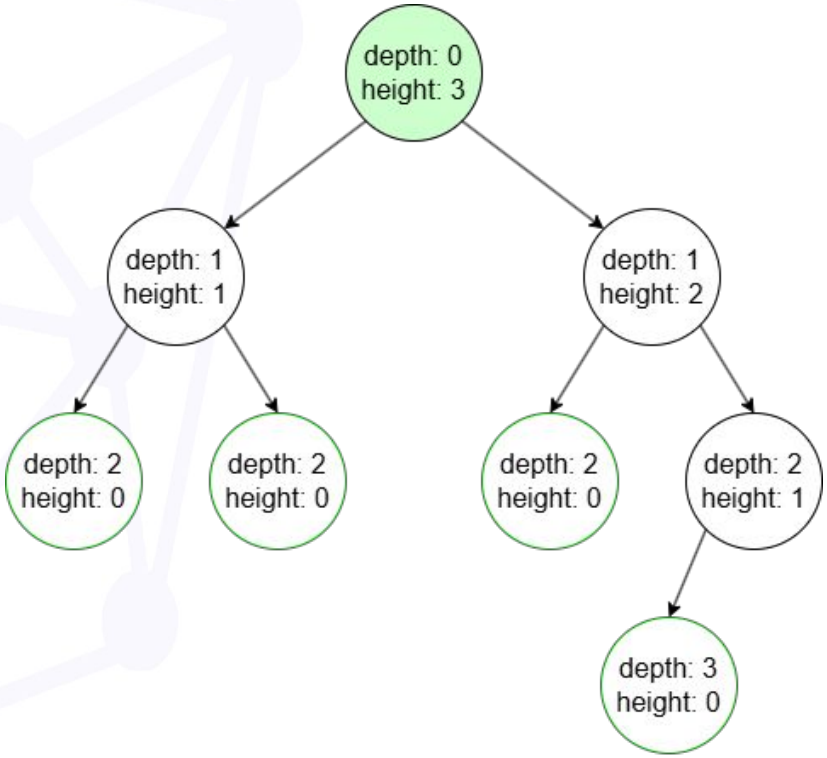


Binarno stablo traženja - BST

- Vremenska složenost osnovnih operacija - $O(h)$
  - h - visina stabla
- Visina stabla u najboljem slučaju je $Log\ n$.
- Visina stabla u najgorem slučaju je n - degenerisano stablo (lista).
- **Balansirano binarno stablo** - binarno stablo traženja sa osobinom da za svaki čvor važi da je razlika visina levog i desnog podstabla što manja - maksimalno 1.
  - Posledično će visina stabla biti $Log\ n$.

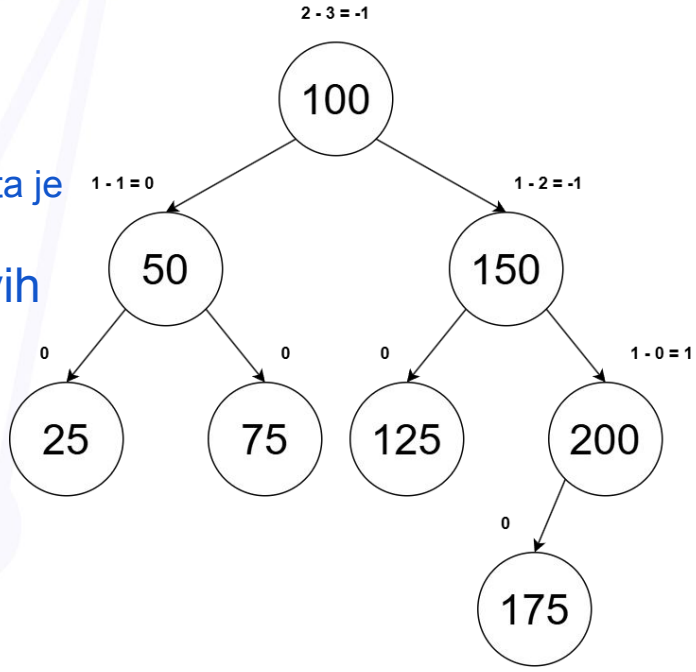
Balansirano stablo traženja - BST

- Koncepti bitni za očuvanje balansiranosti stabla:
  - Dubina čvora:
    - Dužina putanje do korena;
    - Koren ima dubinu 0.
  - Visina stabla:
    - Dužina putanje do najudaljenijeg lista;
    - Listovi imaju visinu 0.



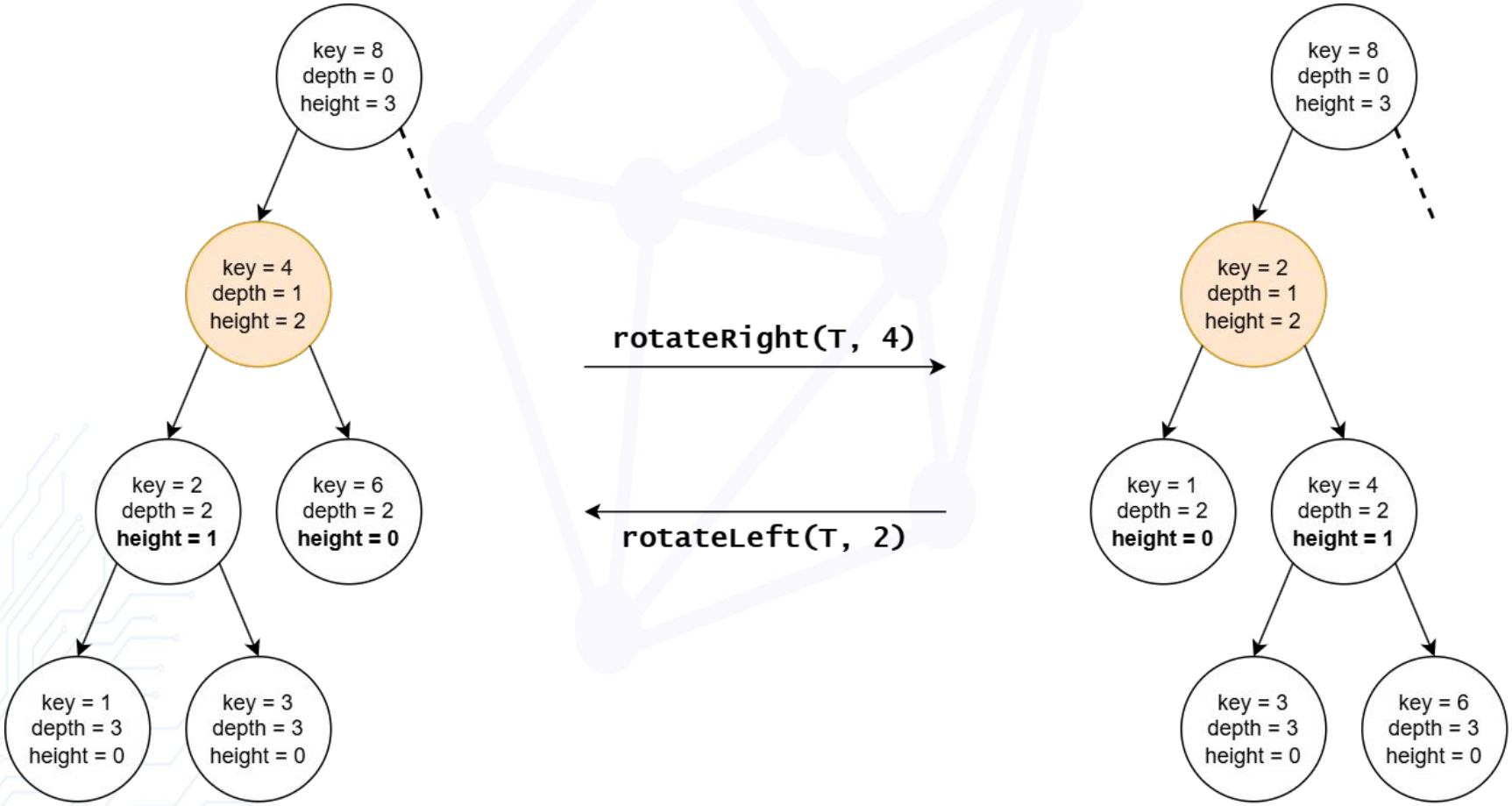
AVL stablo

- Koncept osmišljen 1962. godine od strane sovjetskih naučnika:
  - Georgy **A**delson-**V**elsky;
  - Evgenii **L**andis.
- Oslanja se na koncept binarnog stabla traženja.
- Dodatna osobina - balansiranost:
  - Za svaki čvor važi da razlika visina levog i desnog deteta je maksimalno 1.
- Za svaki čvor se čuva (ili računa) i visina njegovih podstabala.
- Ovakvo stablo je balansirano ⇒ $h < 2*Log\ n$.
- Moguće je narušiti osobine AVL stabla prilikom dodavanja ili brisanja elemenata, stoga je neophodno implementirati mehanizam za **prepoznavanje narušavanja balansiranosti** i **restauraciju balansiranosti**.



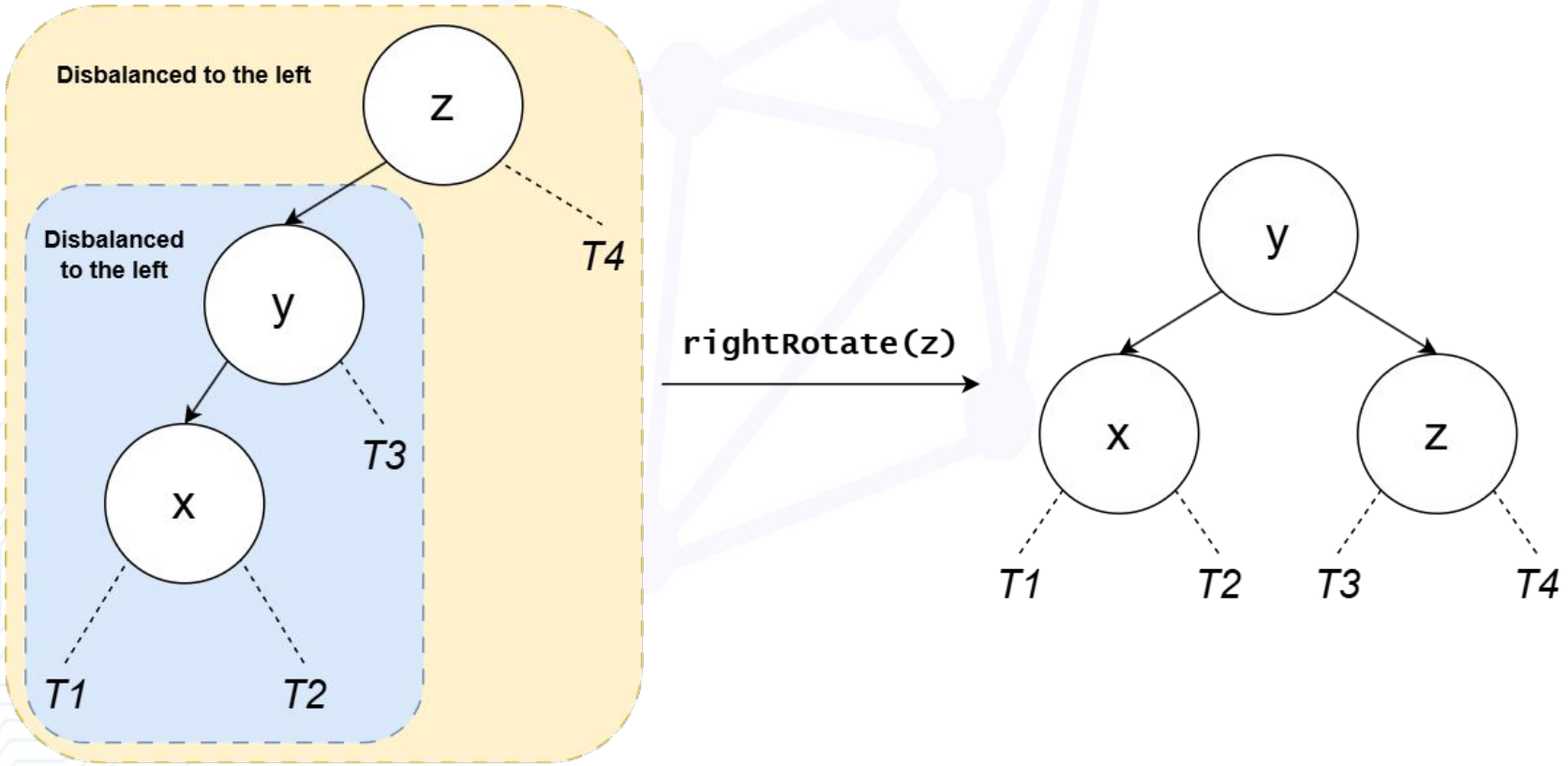
Rotacije - ulevo i udesno

- Rotacijom se visina jednog podstabla smanjuje za 1, a drugog uvećava za 1.



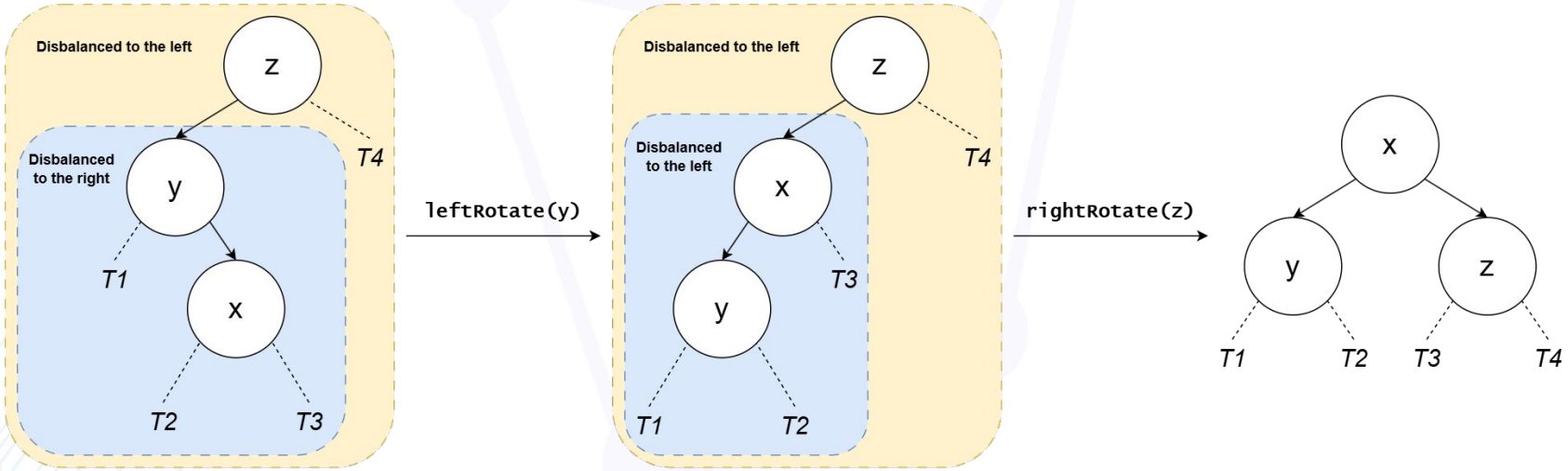
Tipovi neuravnoteženosti - Left Left (LL)

- Za proizvoljni čvor ***z*** uočavamo neuravnoteženost **ulevo**, a njegovo levo podstablo je ponovo neuravnoteženo **ulevo**.
- *Rešenje:* Rotacijom udesno čvora ***z*** dobijamo balansirano stablo.



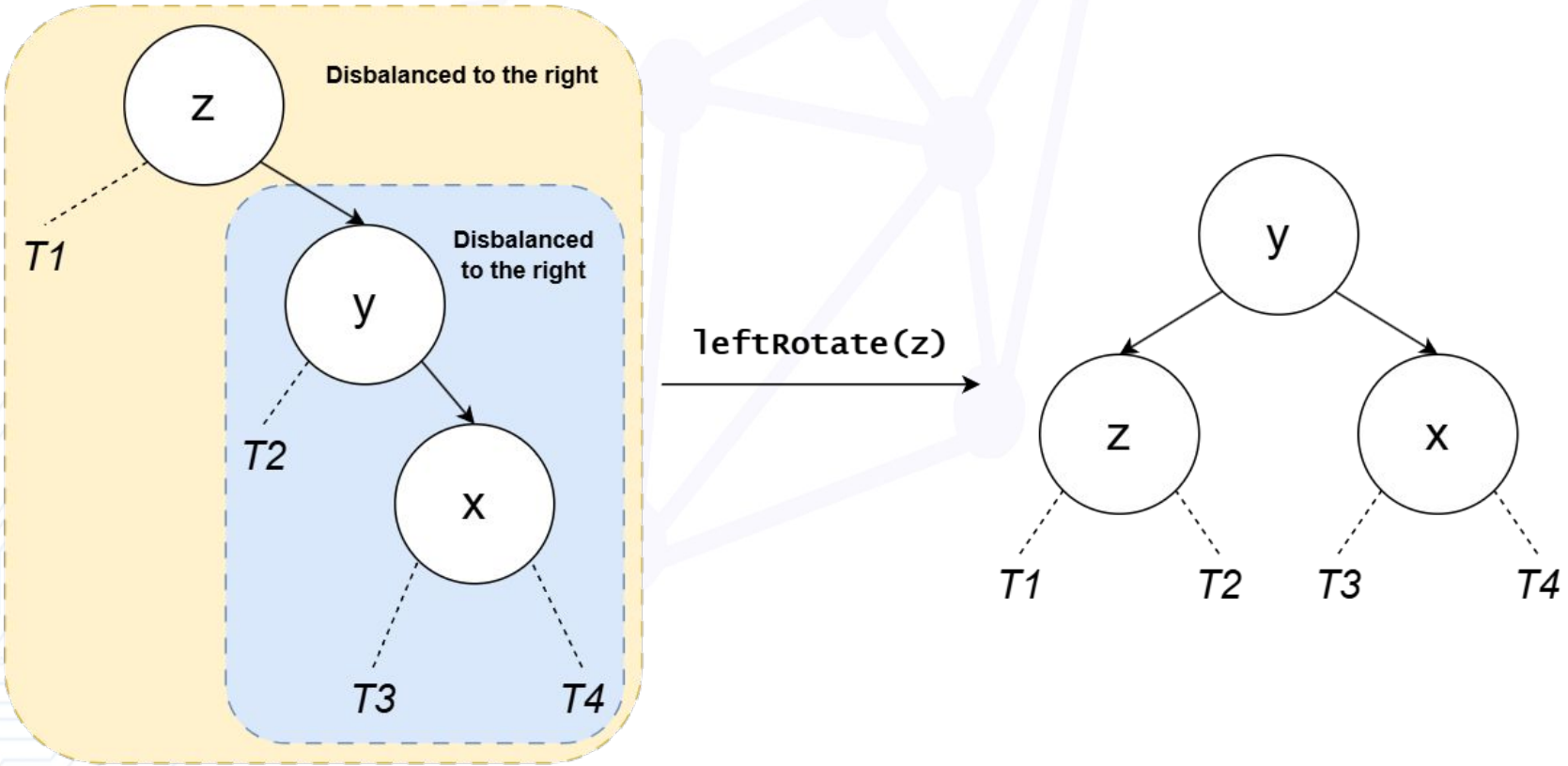
Tipovi neuravnoteženosti - Left Right (LR)

- Za proizvoljni čvor ***z*** uočavamo neuravnoteženost **ulevo**, a njegovo levo podstablo, čiji je koren ***y***, je neuravnoteženo **udesno**.
- *Rešenje:* Rotacijom ulevo čvora ***y*** dobijamo slučaj **LL**, koji rešavamo rotacijom udesno čvora ***z***.



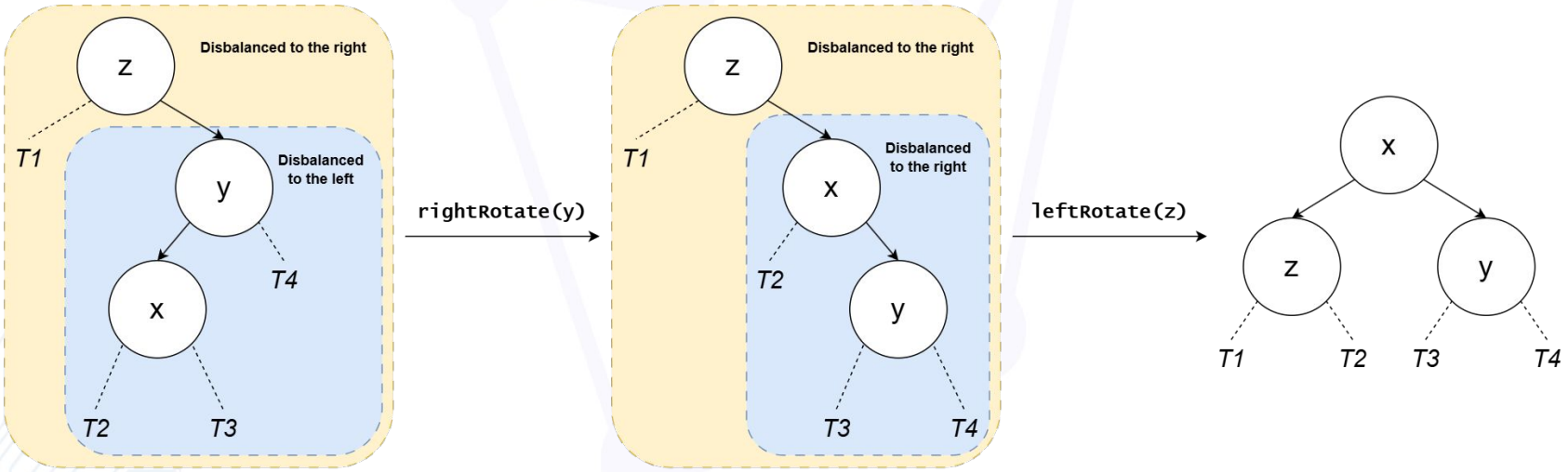
Tipovi neuravnoteženosti - Right Right (RR)

- Za proizvoljni čvor ***z*** uočavamo neuravnoteženost **udesno**, a njegovo desno podstablo je ponovo neuravnoteženo **udesno**.
- *Rešenje:* Rotacijom ulevo čvora ***z*** dobijamo balansirano stablo.



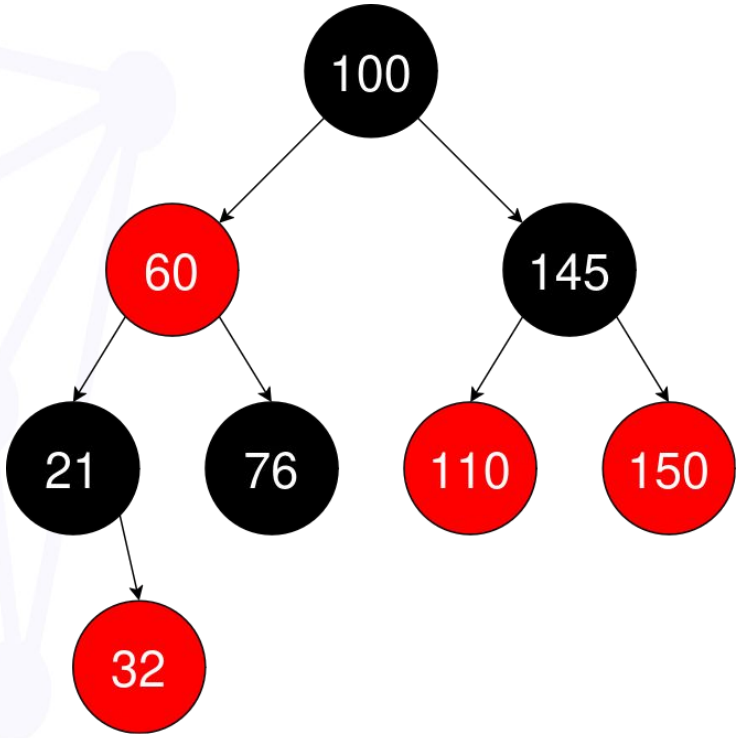
Tipovi neuravnoteženosti - Right Left (RL)

- Za proizvoljni čvor ***z*** uočavamo neuravnoteženost **udesno**, a njegovo desno podstablo, čiji je koren ***y***, je neuravnoteženo **ulevo**.
- *Rešenje:* Rotacijom udesno čvora ***y*** dobijamo slučaj **RR**, koji rešavamo rotacijom ulevo čvora ***z***.



Red-Black Stabla

- Poput AVL stabla, RB stabla su takođe samobalansirajuća binarna stabla pretrage.
- RB stabla se zasnivaju na sledećim pravilima:
  - Svaki čvor je ili crn ili crven;
  - Korenski čvor treba uvek biti crn;
  - Crveni čvor, ne sme imati crveno dete (uzastopni crveni čvorovi).
  - Za proizvoljni čvor, na svakom putu do njegovih listova, mora biti isti broj crnih čvorova.
  - Za potrebe implementacije, nepostojeći čvorovi (npr. oni na koje pokazuju listovi) smatraju se crnim čvorovima.

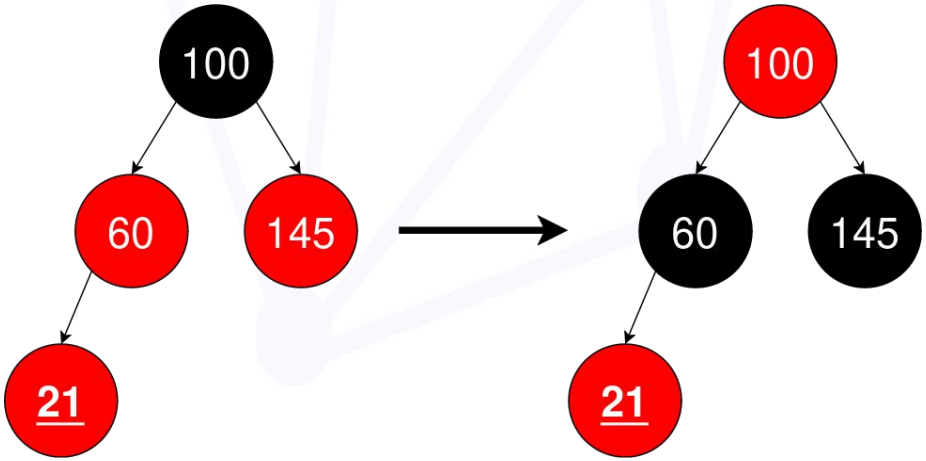


Dodavanje i brisanje čvorova u RB stablu

- Dodavanje je gotovo isto kao i kod AVL stabala, prati se pravilo BST, tj. za proizvoljan čvor i njegov ključ ***K***, svi ključevi u njegovom levom podstablu, moraju biti manji od ***K***, a u desnom podstablu veći od ***K***.
- Novododati čvorovi su uvek **crvene** boje.
- Brisanje elemenata je, takođe, isto kao i kod AVL stabala, međutim bitno je voditi evidenciju o boji obrisanih/pomerenih čvorova:
  - Ako je element koji brišemo bio list ili je imao samo 1 dete - pamtimo boju obrisanog čvora;
  - Ako je element koji brišemo imao 2 deteta (brisanje zamenom) - pamtimo boju naslednika.
- Prilikom dodavanja ili brisanja čvorova, mogu se narušiti pravila RB stabala. U tom slučaju su nam potrebni:
  - Mehanizam za popravljanje RB poretka nakon dodavanja elemenata;
  - Mehanizam za popravljanje RB poretka nakon brisanja elemenata.

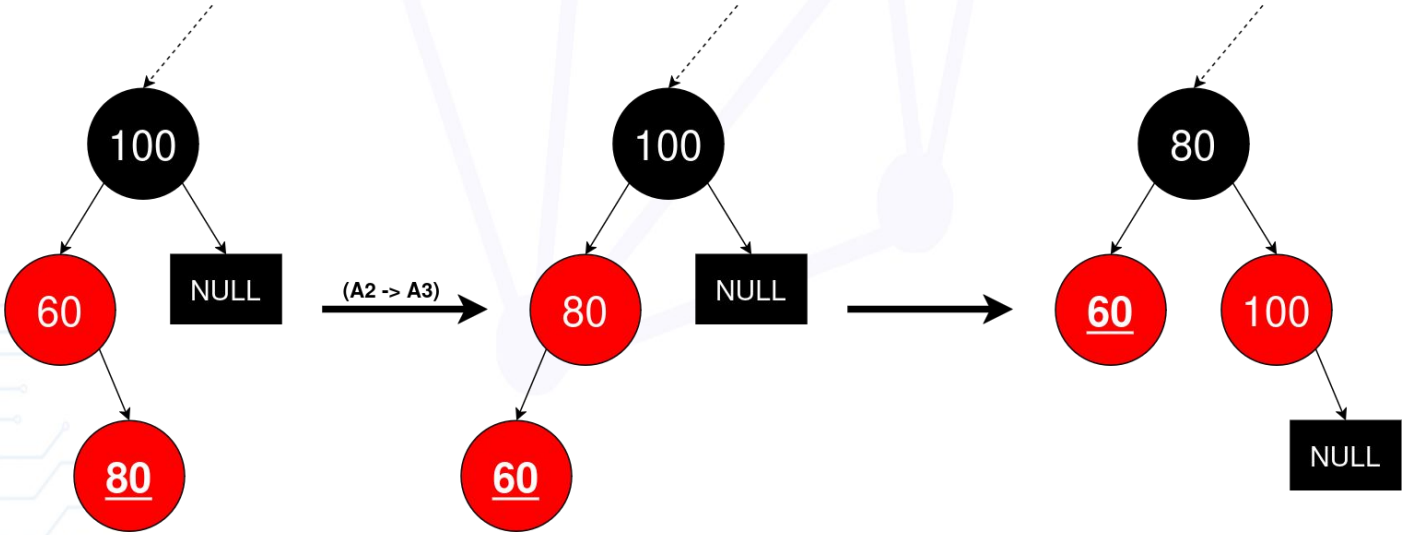
Popravljanje RB poretka nakon dodavanja

- Popravljanje je neophodno samo ukoliko je roditelj novododatog čvora crvene boje.
- Slučaj A - Roditelj je levo dete:
  - Podslučaj A1 - *Uncle* (*sibling* roditeljskog čvora) je crvene boje - sledi promena boja:
    - Roditelj i *Uncle* novododatog čvora postaju crne boje;
    - Roditelj roditelja postaje crvene boje (s obzirom da smo promenili boju elementa na crvenu, potencijalno smo poremetili RB poredak, stoga, ponavljamo ceo postupak nad roditeljem roditelja).



Popravljanje RB poretka nakon dodavanja

- Slučaj A - Roditelj je levo dete:
  - Podslučaj A2 - *Uncle* je crne boje, a novododati čvor je desno dete (cik-cak linija):
    - Rotiramo roditelja novododatog člana ulevo. (nakon rotacije, ponašamo se prema dosadašnjem roditelju novododatog člana, kao da je novododat član i obrnuto)
    - Rotacijom ovo postaje podslučaj A3.
  - Podslučaj A3 - *Uncle* je crne boje, a novododati čvor je levo dete (prava linija):
    - Bojimo roditelja novododatog člana u crno, a roditelja roditelja novododatog člana u crveno.
    - Rotiramo roditelja roditelja novododatog člana udesno.



Popravljanje RB poretka nakon dodavanja

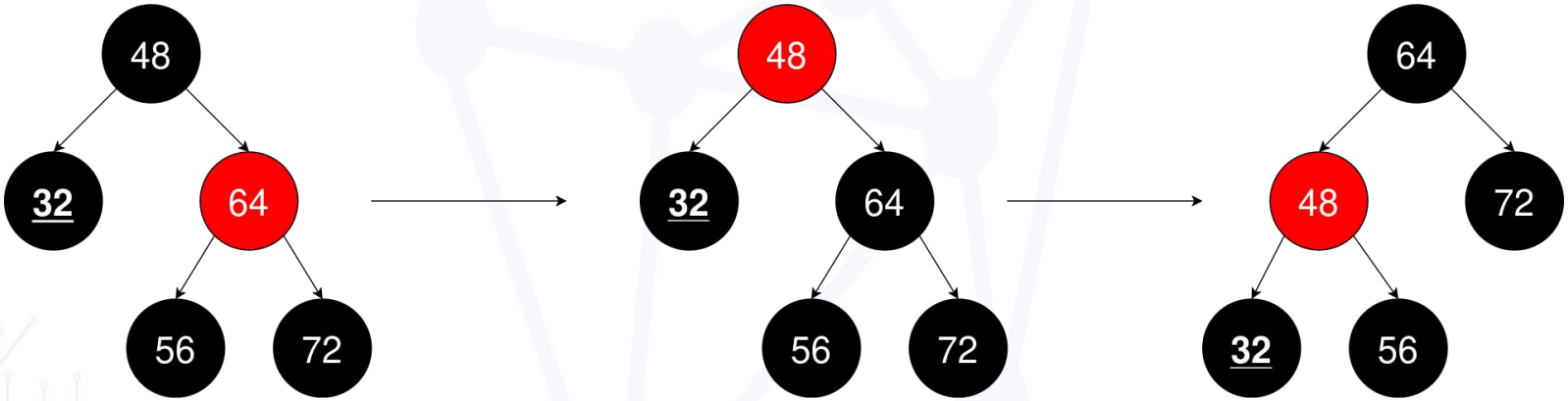
- Slučaj B - Roditelj je desno dete (sve je simetrično sa slučajem A):
  - Podslučaj B1 - *Uncle* (*sibling* roditeljskog čvora) je crvene boje - sledi promena boja:
    - Roditelj i *Uncle* novododatog čvora postaju crne boje;
    - Roditelj roditelja postaje crvene boje (s obzirom da smo promenili boju elementa na crvenu, potencijalno smo poremetili RB poredak, stoga, ponavljamo ceo postupak nad roditeljem roditelja).
  - Podslučaj B2 - *Uncle* je crne boje, a novododati čvor je levo dete (cik-cak linija):
    - Rotiramo roditelja novododatog člana udesno. (nakon rotacije, ponašamo se prema dosadašnjem roditelju novododatog člana, kao da je novododat član i obrnuto)
    - Rotacijom ovo postaje podslučaj B3.
  - Podslučaj B3 - *Uncle* je crne boje, a novododati čvor je desno dete (prava linija):
    - Bojimo roditelja novododatog člana u crno, a roditelja roditelja novododatog člana u crveno.
    - Rotiramo roditelja roditelja novododatog člana ulevo.

Popravljanje RB poretka nakon brisanja

- Popravka je neophodna samo u slučaju da je obrisan čvor bio crne boje.
- Potencijalni problem - S obzirom na to da smo obrisali crni čvor, verovatno bar jedna od putanja sada ima jedan crni čvor manje na putu do listova.
- Ako je čvor, koji je zamenio obrisani crni čvor, crven, rešenje je vrlo jednostavno, samo ga obojimo u crno.
- Ako je čvor, koji je zamenio obrisani crni čvor, crn, onda je rešenje kompleksnije i raspoznajemo sledeće slučajeve:
  - A - Čvor koji je zamenio obrisani čvor je levo dete:
    - A1 - *Sibling* zamene je crvene boje;
    - A2 - *Sibling* zamene je crne boje i oba njegova deteta su crne boje;
    - A3 - *Sibling* zamene je crne boje, njegovo levo dete je crvene, a desno crne boje;
    - A4 - *Sibling* zamene je crne boje i njegovo desno dete je crvene boje;
  - B - Čvor koji je zamenio obrisani čvor je desno dete:
    - Podslučajevi B1-4 odgovaraju A1-4 samo su simetrični.

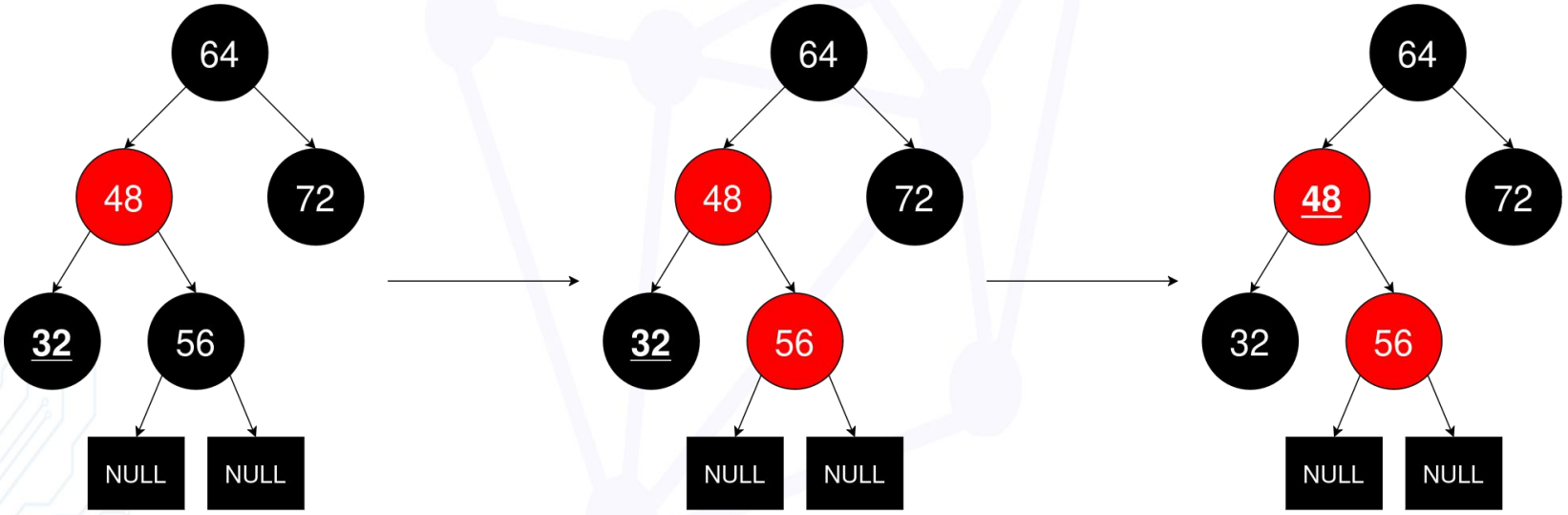
Popravljanje RB poretka nakon brisanja

- A1 - *Sibling* zamene (***S***) je crvene boje:
  - Menjamo boju ***S*** sa njegovim roditeljem;
  - Rotiramo ulevo roditelja od ***S***;
  - Ovim promenama nismo rešili problem, ali smo transformisali slučaj u A2, A3 ili A4.



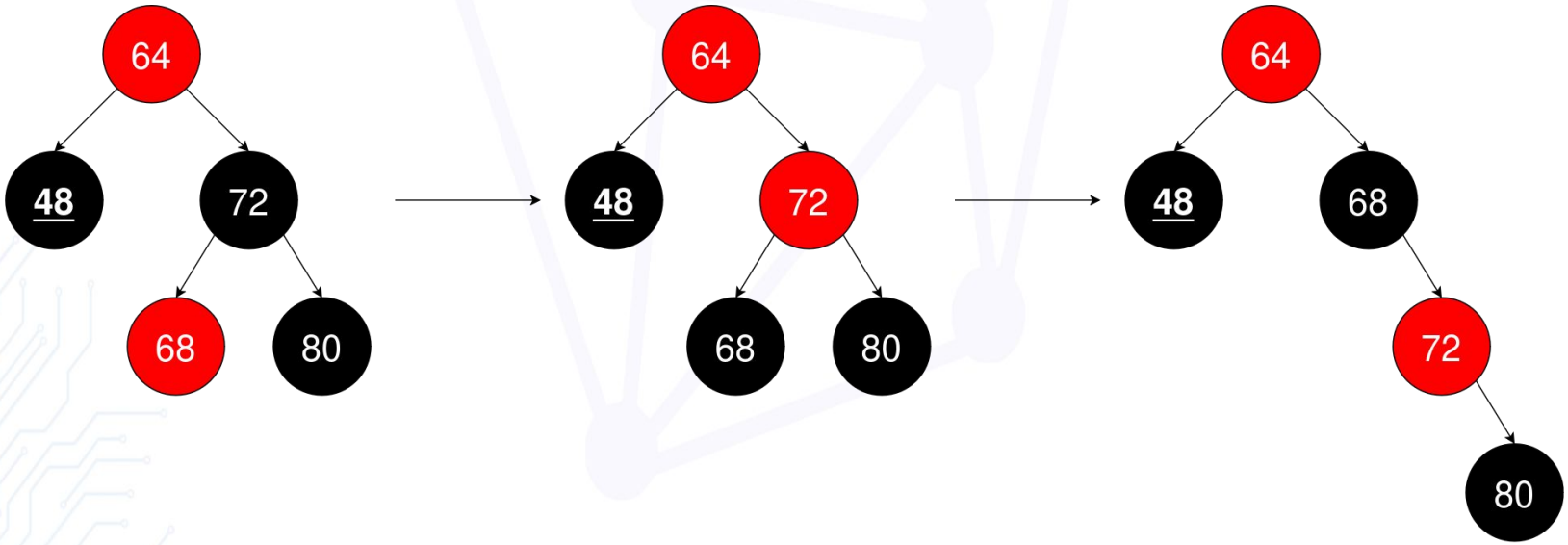
Popravljanje RB poretka nakon brisanja

- A2 - *Sibling* zamene (***S***) je crne boje i oba njegova deteta su crne boje:
  - Menjamo boju ***S*** u crvenu;
  - Ponavljamo ceo postupak **Popravke RB poretka nakon brisanja** za roditelja od ***S***.



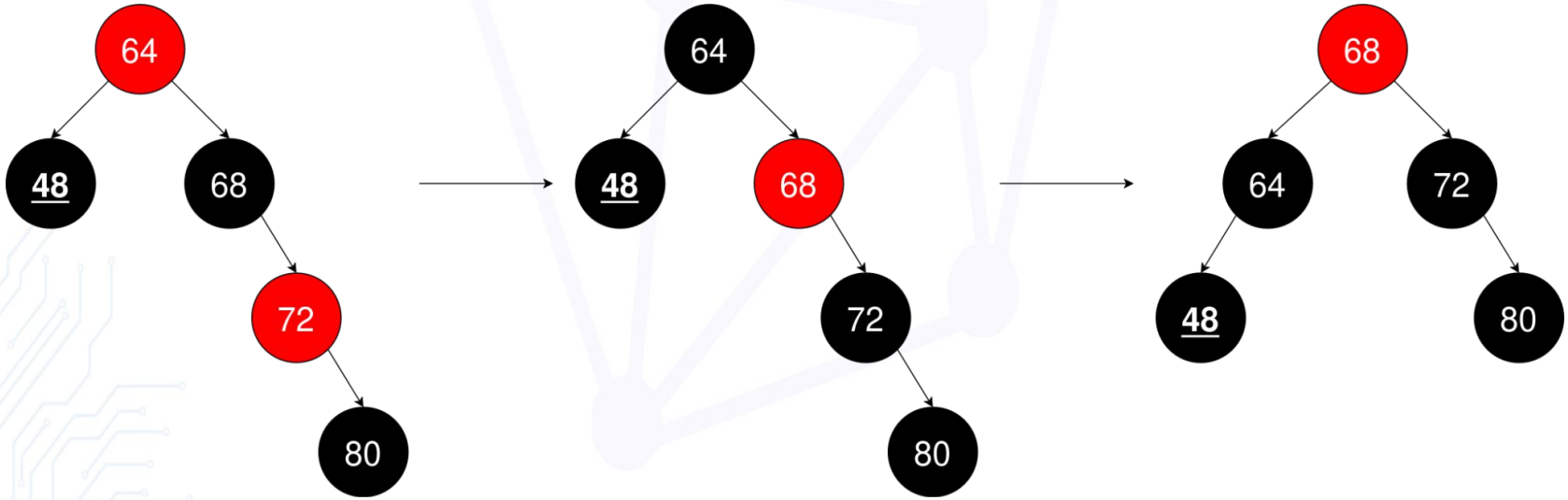
Popravljanje RB poretka nakon brisanja

- A3 - *Sibling* zamene (***S***) je crne boje, njegovo levo dete je crvene, a desno crne boje:
  - Menjamo boje čvora ***S*** i njegovog levog deteta;
  - Čvor ***S*** rotiramo udesno.
  - Ovim promenama smo namestili da je novi *Sibling* zamene crne boje i da ima desno dete crvene boje, što nas vodi u slučaj A4.



Popravljanje RB poretka nakon brisanja

- A4 - *Sibling* zamene (***S***) je crne boje i njegovo desno dete je crvene boje:
  - Čvoru ***S*** dodeljujemo boju njegovog roditelja;
  - Roditelju i desnom detetu čvora ***S*** dodeljujemo crnu boju;
  - Roditelja čvora ***S*** rotiramo ulevo;
  - Kada se izvrši postupak namenjen za slučaj A4, možemo da konstatujemo da je uspešno izvršena popravka RB poretka.



RB vs AVL stabla

- Oba pristupa garantuju da će operacije pretrage, dodavanja i brisanja biti izvršene u vremenu ***O(Log n)***.
- AVL stabla su “jače” balansirana, što rezultuje manjom ukupnom visinom stabla.
- RB stabla su “labavije” balansirana, ali i dalje operacije ostaju u okvirima ***O(Log n)***.
- Iako iz prethodnih konstatacija AVL stabla deluju kao bolja opcija, u praksi zapravo AVL stabla, u odnosu na RB stabla, rade dosta više rotacija prilikom dodavanja i brisanja, iz čega možemo zaključiti:
  - AVL Stabla su bolja opcija kada nam je potrebno efikasno čitanje;
  - RB Stabla su bolja opcija kada nam je potrebno efikasno dodavanje i brisanje elemenata;