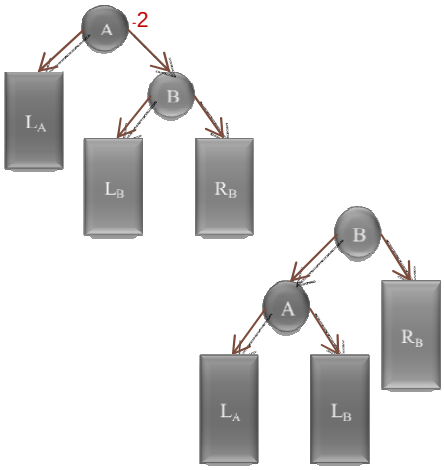
ALGORYTMY I STRUKTURY DANYCH –DRZEWO AVL

- **Drzewo AVL –** binarne drzewo wyszukiwań, w którym:
  - dla każdego węzła y należącego do drzewa współczynnik wyważenia bf(y) $\in$ {-1,0,1} (tj. wysokość lewego ($h_L(y)$) i prawego ($h_R(y)$) poddrzewa y różni się co najwyżej o jeden)

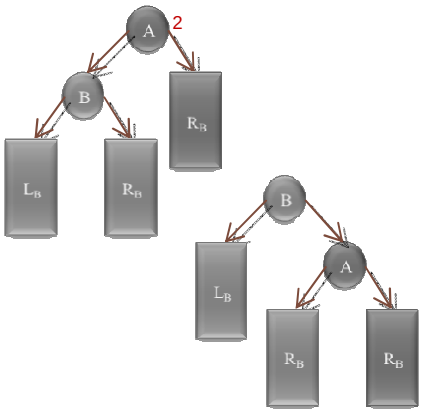
- **Operacje na drzewie AVL**
  - takie same jak dla drzewa BST
  - rotacje (przywraca wyważenie)
  - operacje dodawania i usuwanie węzła wymagają aktualizacji współczynnika bf i mogą wymagać rotacji

  - rotacja pojedyncza w prawo (RR)
    - **dane**: drzewo AVL, węzły A (zaburza zrównoważenie), B (=right[A]) i P (=p[A])
    - **wynik**: węzeł AVL

right[A] ← left[B]
if (right[A] ≠ NIL)
  p[right[A]] ← A
left[B] ← A, p[B] ← P, p[A] ← B,
if (P ≠ NIL)
  if (left[P] = A) left[P] ← B
  else right[P] ← B
else
  root ← B;
if(bf[B] = -1) bf[A] ← bf[B] ← 0
else (bf[B] = 0) bf[A] ← -1, bf[B] ← 1

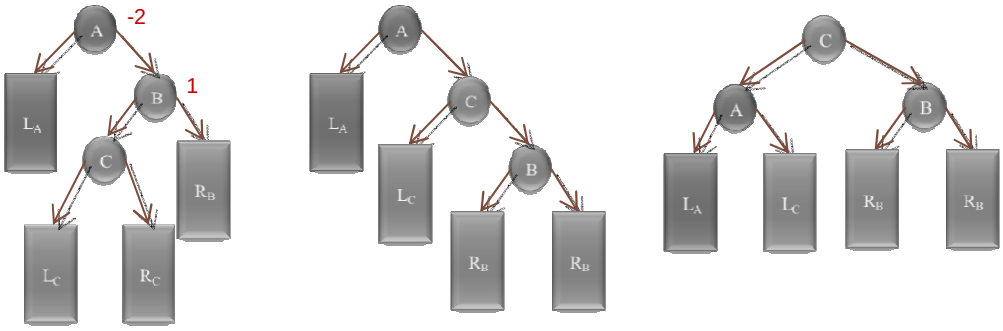


  - rotacja pojedyncza w lewo (LL)



if(bf[B] = 1) bf[A] ← bf[B] ← 0
else (bf[B] = 0) bf[A] ← 1, bf[B] ← -1

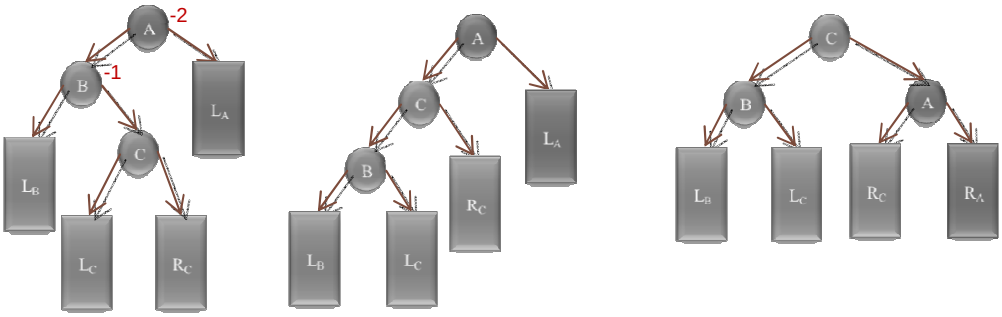
+ rotacja podwójna w prawo – lewo (RL)



LL + RR

bf[C] = -1 → bf[A] = 1, bf[B] = 0, bf[C] = 0
bf[C] = 0 → bf[A] = 0, bf[B] = 0, bf[C] = 0
bf[C] = 1 → bf[A] = 0, bf[B] = -1, bf[C] = 0

+ rotacja podwójna w lewo – prawo (LR)



RR + LL

bf[C] = -1 → bf[A] = 0, bf[B] = 1, bf[C] = 0
bf[C] = 0 → bf[A] = 0, bf[B] = 0, bf[C] = 0
bf[C] = 1 → bf[A] = -1, bf[B] = 0, bf[C] = 0