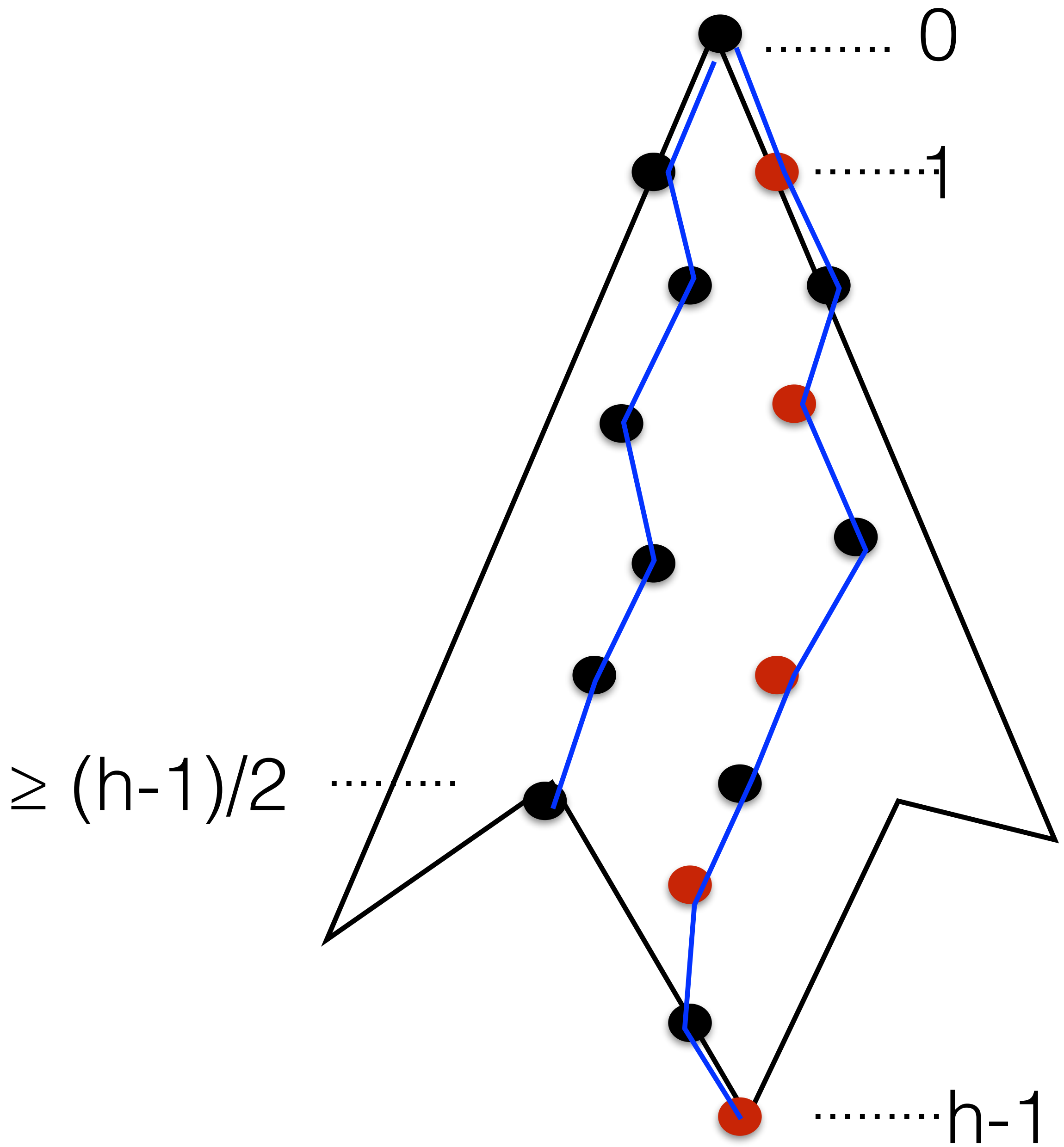
Drzewa czerwono-czarne

Definicja *Drzewo czerwono-czarne* to drzewo poszukiwań binarnych, które spełnia następujące warunki

1. każdy węzeł mam przypisany kolor: czerwony lub czarny
2. korzeń jest czarny
3. liście są czarne; przyjmujemy wariant drzewa z wartownikami - liśćmi są wartownicy
4. czerwony węzeł nie może mieć czerwonego syna
5. na każdej ścieżce od korzenia do liści jest tyle samo czarnych węzłów

Stwierdzenie Drzewo czerwono-czarne posiadające *n* węzłów wewnętrznych ma wysokość nie większą niż $2 \cdot lg(n+1)$

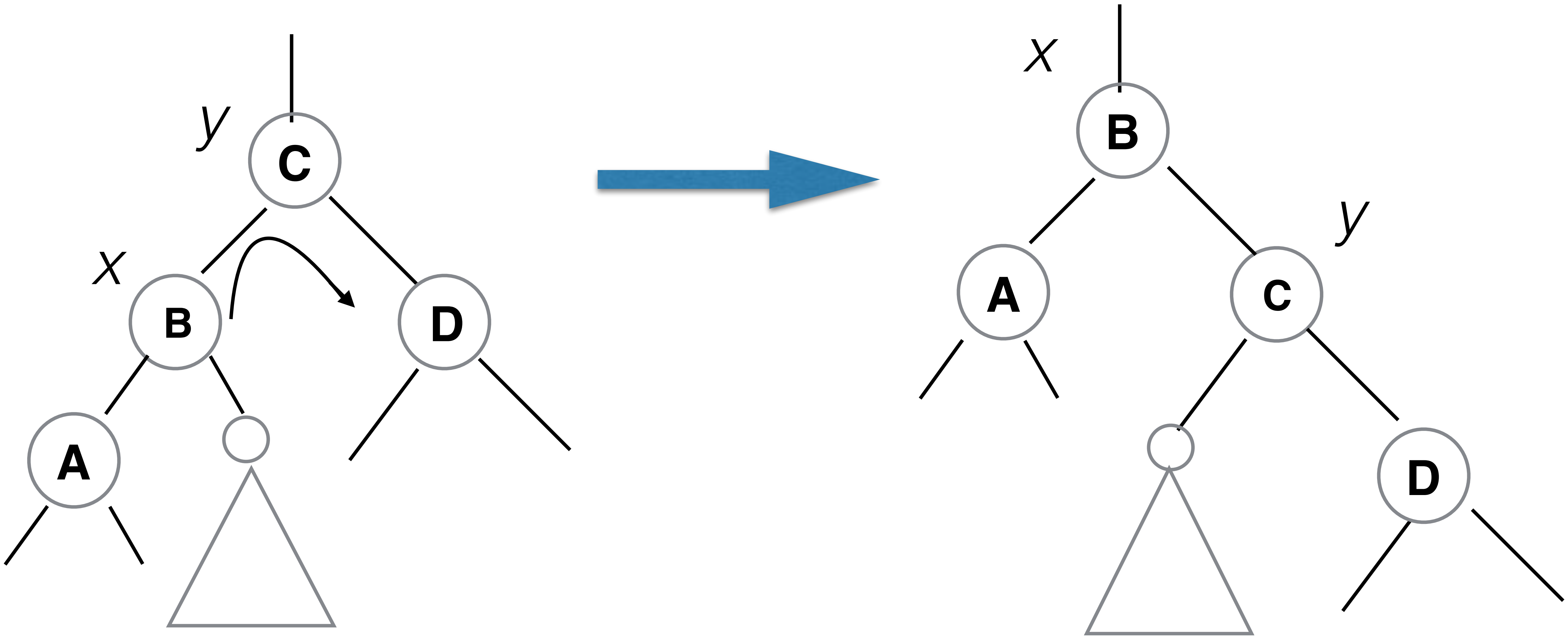


$$n \geq 2^{(h-1)/2+1} - 1 \geq 2^{h/2} - 1$$

$$lg(n+1) \geq h/2$$

$$2\, lg(n+1) \geq h$$

Rotacja w prawo wokół wokół węzła *y*



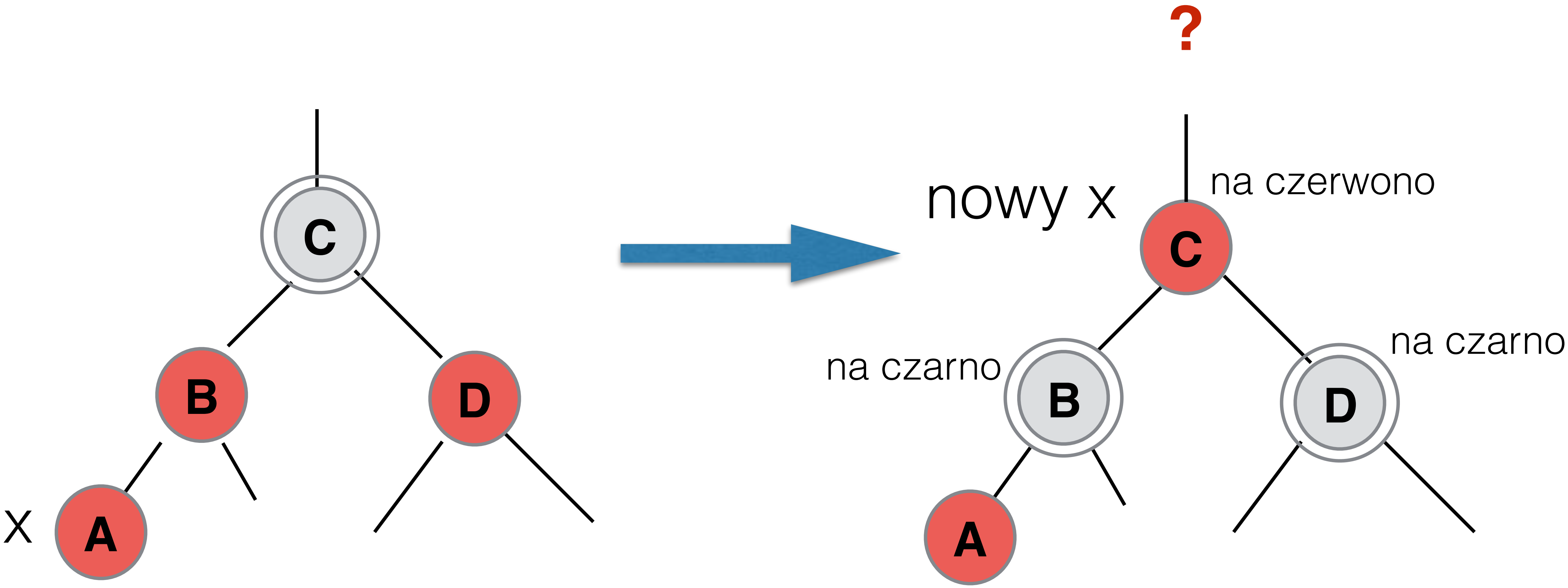
Rotacja w lewo — symetrycznie

Własność: rotacje zachowują uporządkowanie drzewa poszukiwań binarnych

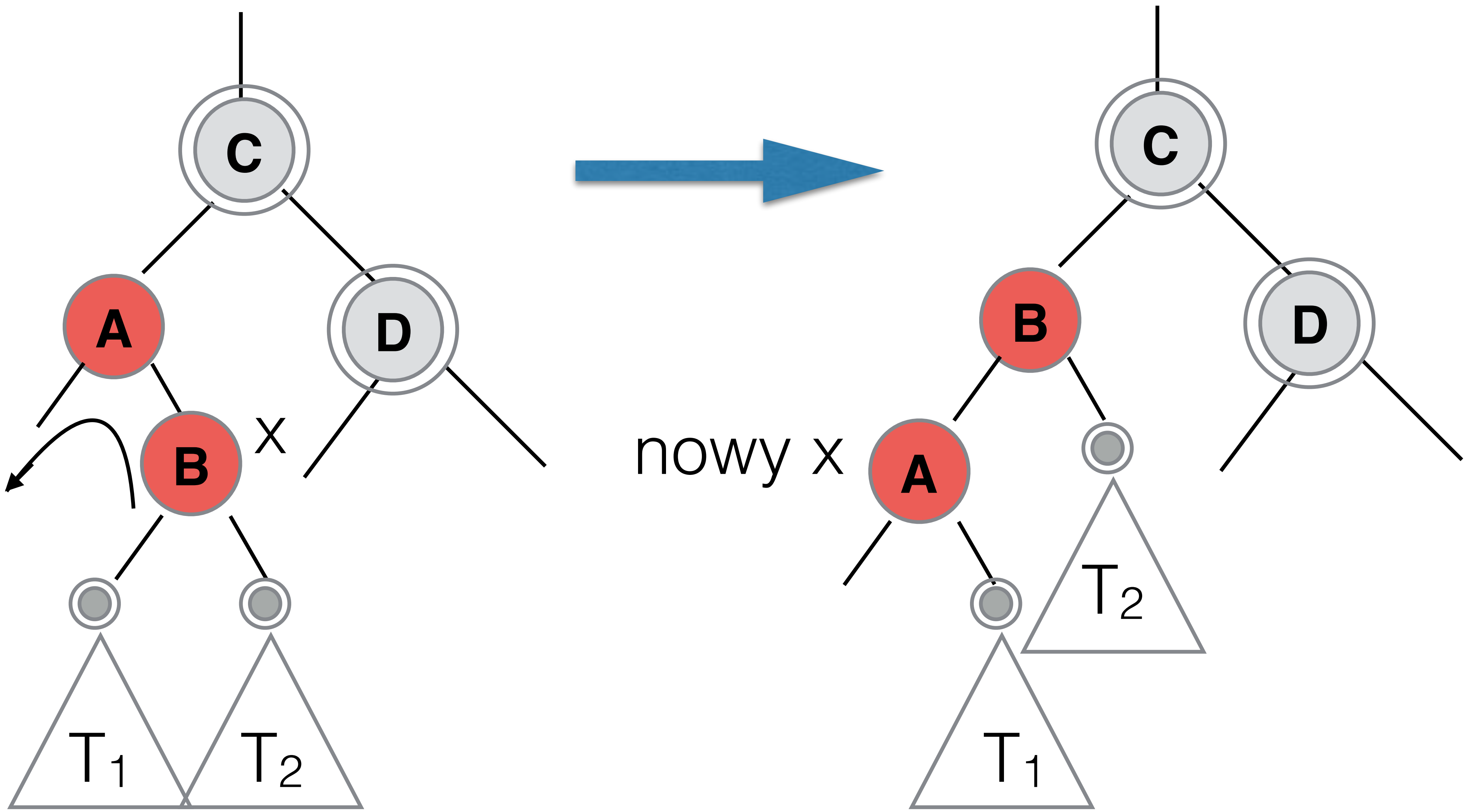
Drzewa czerwono-czarne przypadki wstawiania

x oznacza czerwony węzeł, którego ojciec może być czerwony.
Jeżeli tak jest, dopasowujemy któryś z poniższych przypadków.

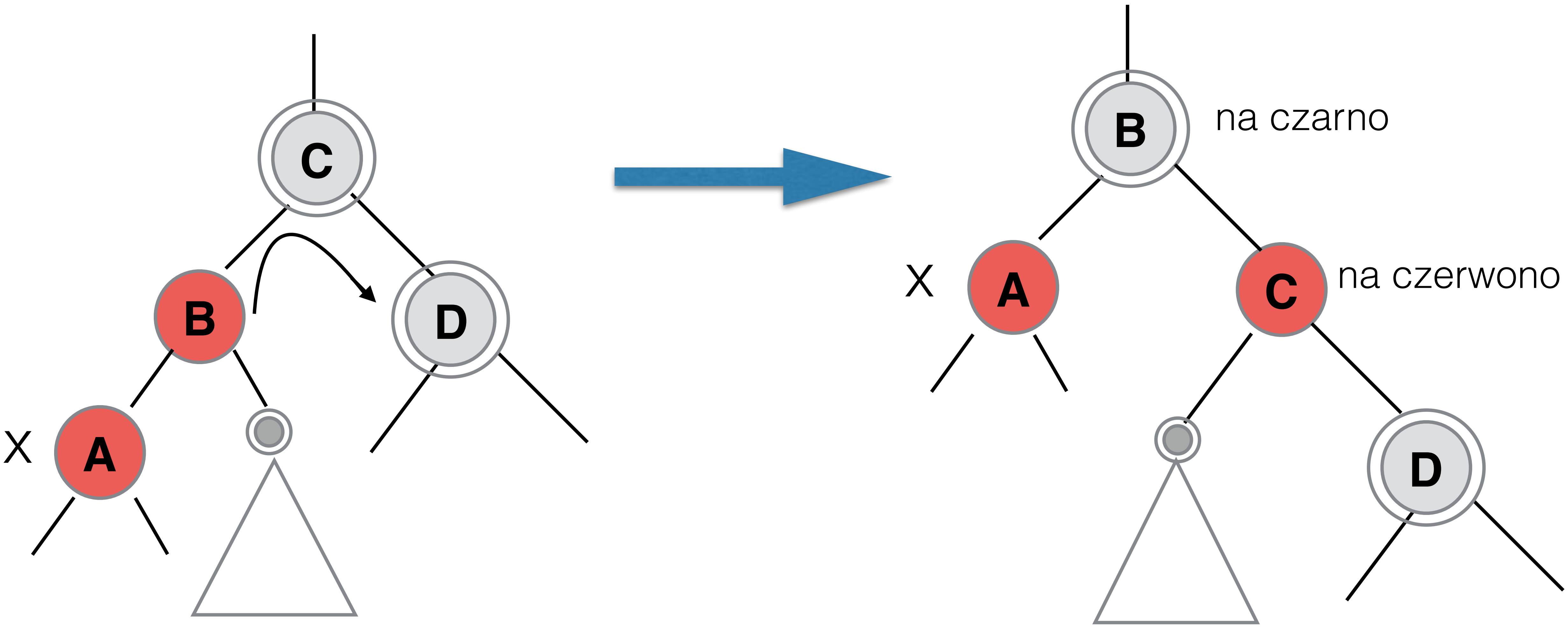
Uwaga: czarne węzły mogą być wartownikami



Przypadek 1: brat ojca x czerwony.
(x może być lewym - jak na rysunku - albo prawym synem swojego ojca)



Przypadek 2: brat ojca węzła x jest czarny, węzeł x i jego ojciec leżą w różnych kierunkach tworząc zakręt. Doprowadzamy do przypadku 3.



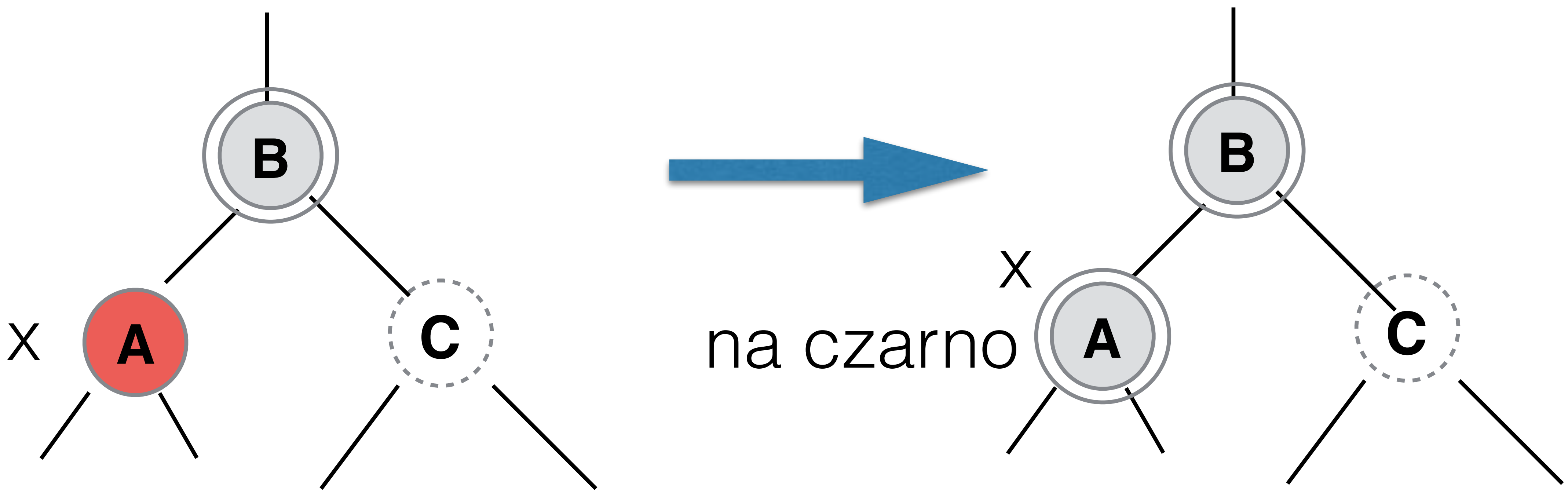
Przypadek 3: brat ojca węzła x jest czarny, węzeł x i jego ojciec leżą w tym samym kierunku. Drzewo naprawione.

```
RB-insert(T,x)
  BST-insert(T,x) // wersja z wartownikiem
  x.color=RED
  while x!=T.root and x.p.color==RED
      if x.p == x.p.p.left  // x.p jest lewym synem
           y = x.p.p.right  // y = brat ojca
           if y.color == RED    // przypadek 1
              x.p.color = BLACK
              y.color = BLACK
              x.p.p.color = RED
              x = x.p.p    // nowy x
            else if x == x.p.right   // przypadek 2
                      x = x.p
                      Left-rotate(T,x)
                  x.p.color = BLACK  // przypadek 3
                  x.p.p.color = RED
                  Right-rotate(T, x.p.p)
      else
          // symetrycznie
          // . . . . . . .
      T.root.color = BLACK
```

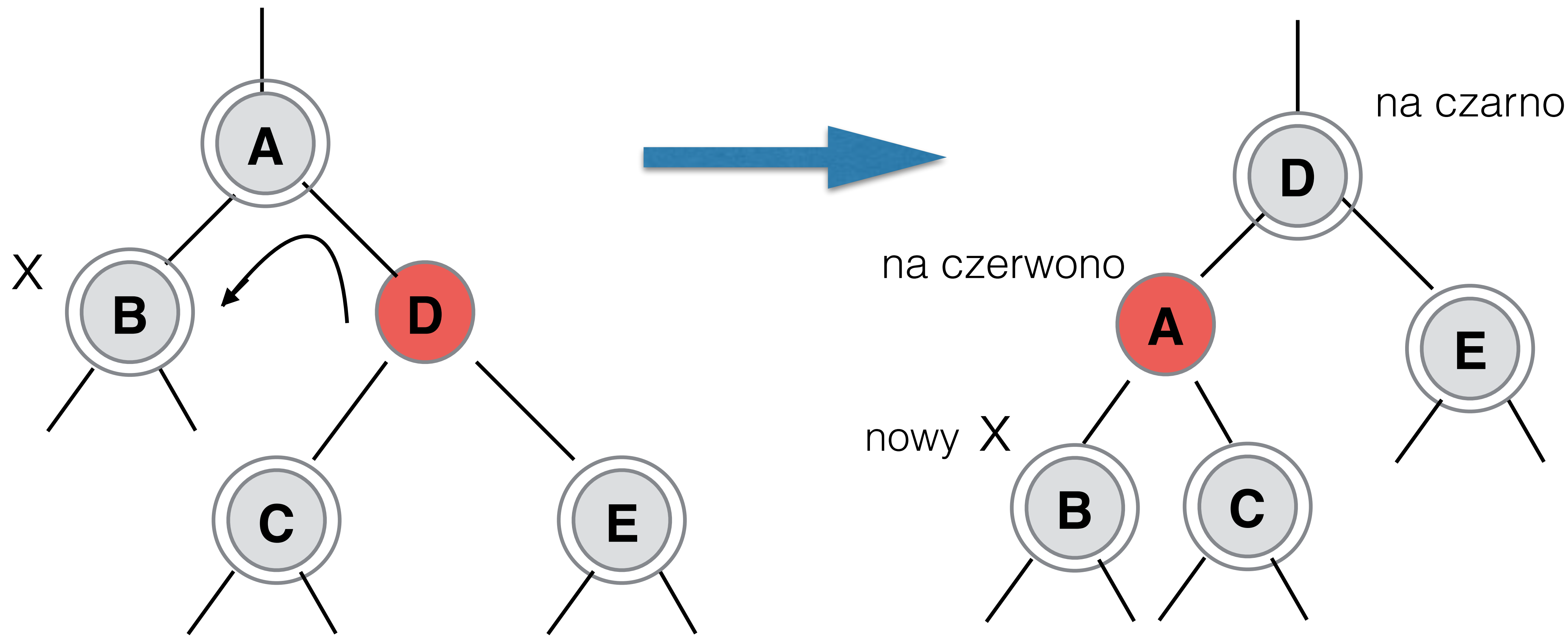
Drzewa czerwono-czarne przypadki usuwania

x oznacza węzeł, od którego zaczynając brakuje jednego czarnego węzła na ścieżkach w dół drzewa

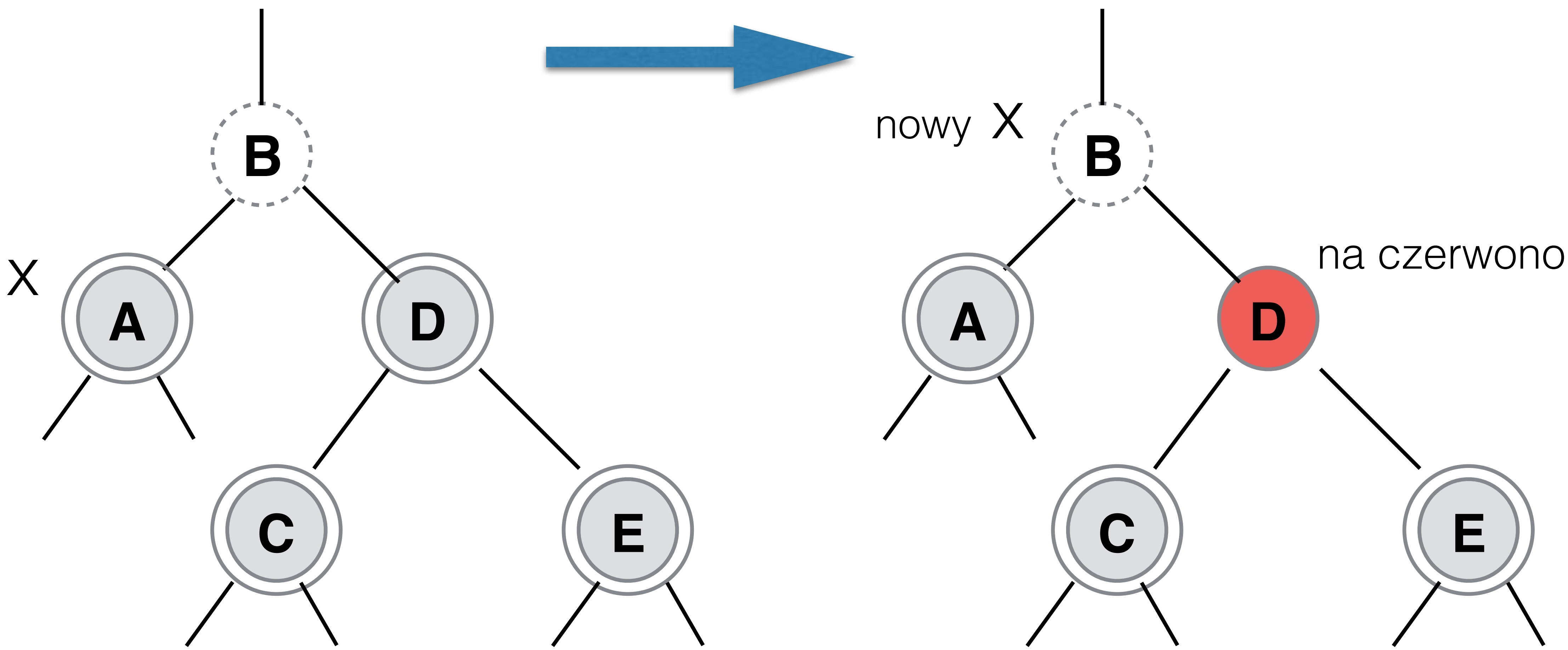
zarówno x jak i inne czarne węzły mogą być wartownikami



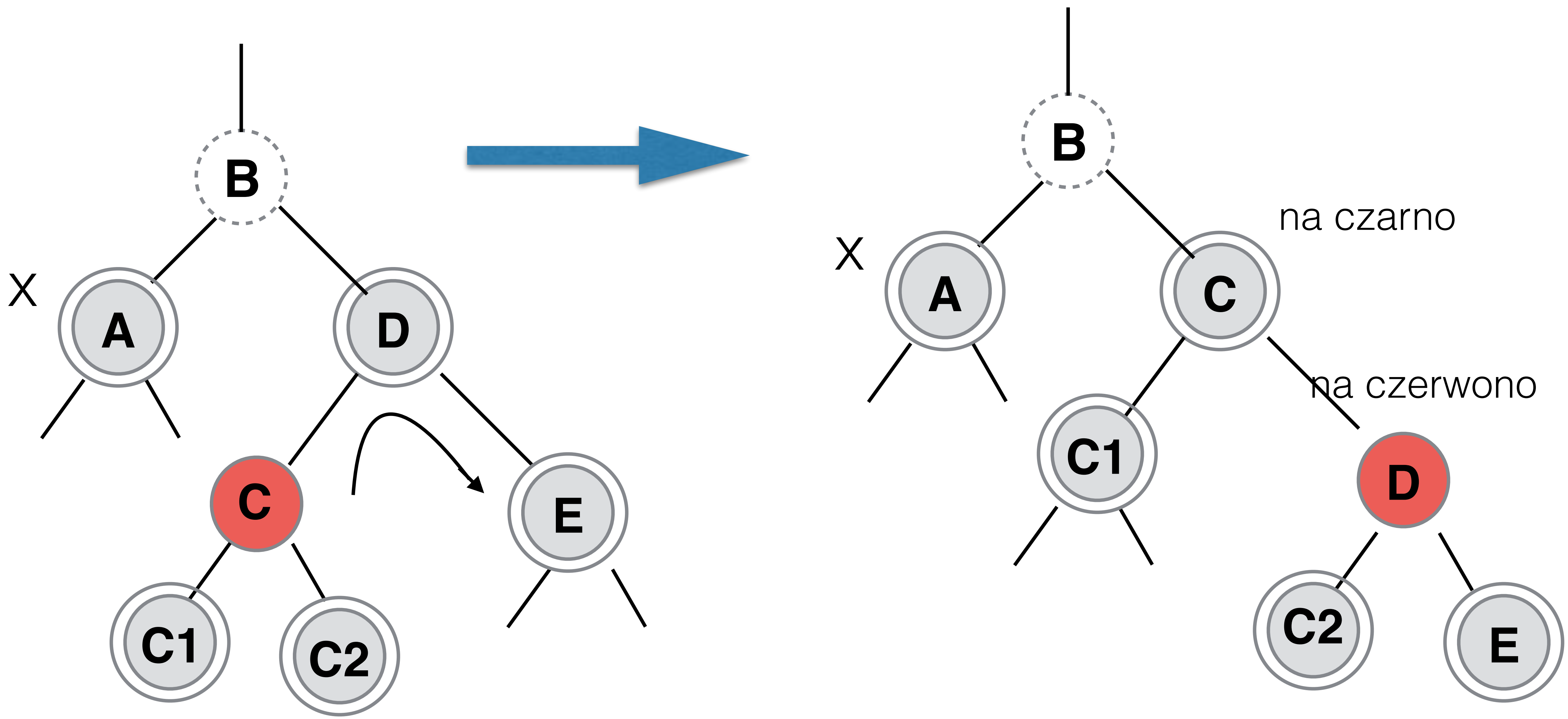
Przypadek 0: węzeł x jest czerwony



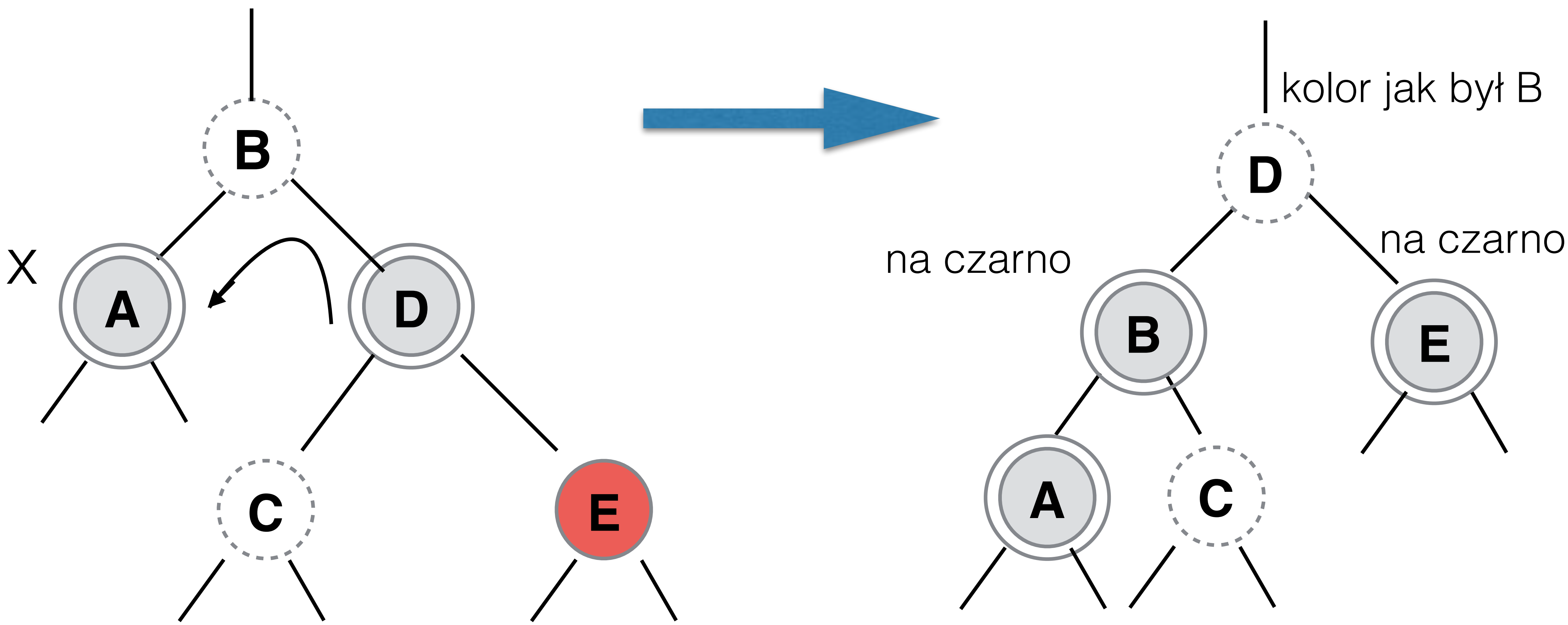
Przypadek 1: brat x czerwony



Przypadek 2: brat węzłą x i obydwaj synowie brata czarni



Przypadek 3: brat x czarny, syn brata skierowany tak jak x jest czerwony a drugi syn czarny; doprowadzamy do przypadku 4



Przypadek 4: brat x czarny, syn brata skierowany przeciwnie niż x jest czerwony. Drzewo naprawione.

Stwierdzenie Przedstawione algorytmy wstawiania i usuwania węzła w drzewie czerwono-czarnym są poprawne, tzn. po wykonaniu każdej z tych operacji drzewo nadal spełnia warunki wymagane od drzewa czerwono-czarnego.

Stwierdzenie Operacje wstawiania, szukania i usuwania wykonywane na drzewie czerwono-czarnym mają czas pesymistyczny $\Theta(lg\ n)$.