

Red-Black stromy

Jiří Zacpal



KATEDRA INFORMATIKY
UNIVERZITA PALACKÉHO V OLOMOUCI

KMI/ZADS - Základní algoritmy a datové struktury

Red-black stromy



- Do vrcholů přidáme položku color, která může nabývat hodnot red a black.
- Místo hodnoty nil použijeme speciální vrchol NIL. Všechny listy jsou tak NIL vrcholy.
- Dále definujeme podmínku vymezující red-black strom.
 1. Kořen je černý.
 2. NIL je černý.
 3. Pokud je vrchol červený, oba jeho potomci jsou černí.
 4. Pro každý vrchol platí, že všechny cesty z něj do listů obsahují stejný počet černých vrcholů.

Uzel



```
{  
    "key":None,  
    "left":None,  
    "right":None,  
    "parent":None,  
    "c":  
}
```

```
rb_tree={"root":{"c":BLACK,"key":None,"parent":None}}
```

Vkládání

Vložení uzlu do stromu

```
def rb_insert(t, added):  
    added["left"] = {"c": BLACK, "key": None, "parent": added}  
    added["right"] = {"c": BLACK, "key": None, "parent": added}  
    added["c"] = RED  
    tree_insert(t, added)  
    rb_fixup(t, added)  
  
tree = {"root": {"c": BLACK, "key": None, "parent": None}}  
u = {"parent": None, "key": 1}
```

Kontrola stromu

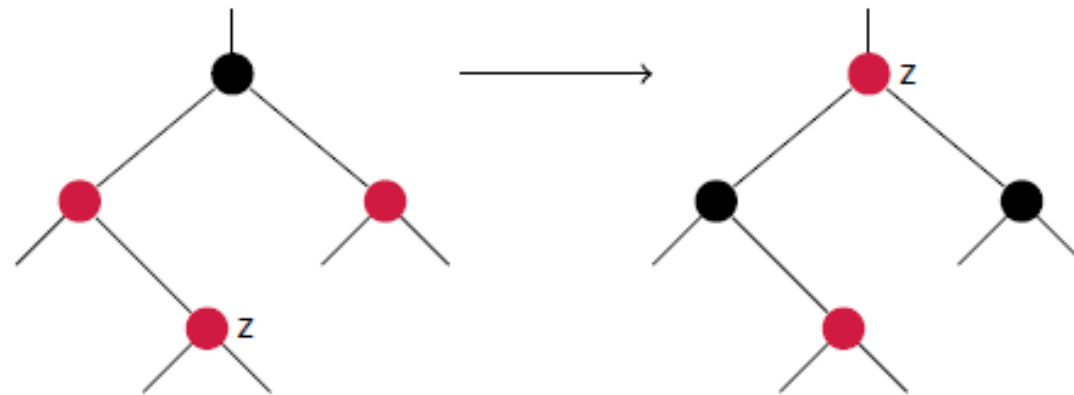


```
def rb_fixup(t,z):  
    while (z != t["root"]):  
        if z["parent"]["c"] == BLACK:  
            break #algorithmus končí  
        z = local_fix(t,z) # procedura pro opravy, viz dalsi slajdy  
t["root"]["c"] = BLACK
```

Úpravy stromu



```
def local_fix(t,z):  
    u=uncle(z)  
    if u["c"]==RED:  
        z["parent"]["c"]=BLACK  
        z["parent"]["parent"]["c"]=RED  
        u["c"]=BLACK  
    return z["parent"]["parent"]
```

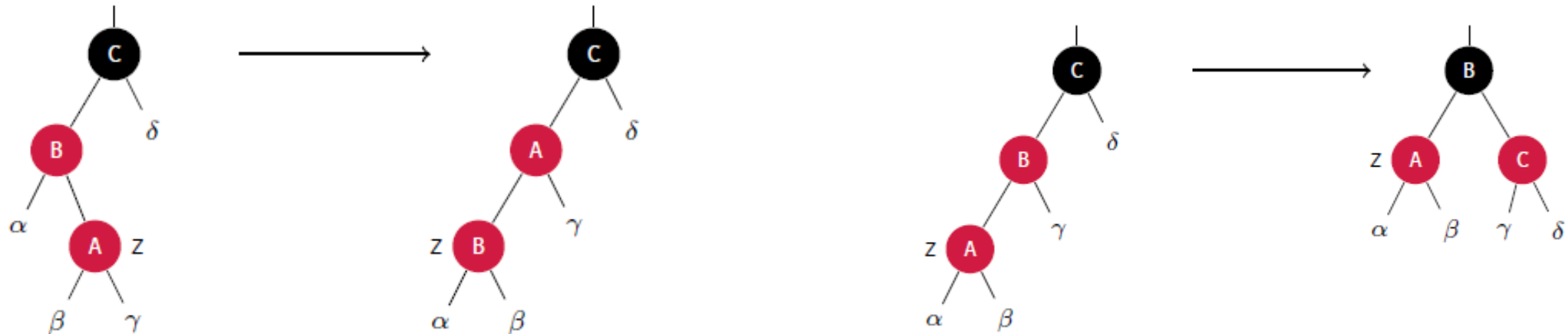


Úpravy stromu



else:

```
if(z["parent"]["parent"]["left"]==z["parent"]):  
    if z["parent"]["right"]["key"]==z["key"]:  
        rotate_left(t,z["parent"])  
    z["parent"]["c"]=BLACK  
    z["parent"]["parent"]["c"]=RED  
    rotate_right(t,z["parent"]["parent"])
```



Úpravy stromu



```
else:
    if z["parent"]["left"]==z:
        rotate_right(t,z["parent"])
    z["parent"]["c"]=BLACK
    z["parent"]["parent"]["c"]=RED

    rotate_left(t,z["parent"]["parent"])
return z["parent"]
```

1. Použijte RB strom pro ukládání počtu bodů u studentů.
2. Napište funkce:
 - `pridej_body(s, jmeno, body)` – která vloží do stromu osobu se jménem `jmeno` a body `body`. Pokud tato osoba ve stromu již bude, tak mu body přidá.
 - `body_celkem(s, jmeno)` – která pro daného studenta počet získaných bodů, případně informaci, že student není v seznamu.
- Příklad:
 - Příkazy:

```
pridej_body(t, "Pavel", 2)
pridej_body(t, "Jirka", 1)
pridej_body(t, "Alena", 3)
pridej_body(t, "Pavel", 4)
pridej_body(t, "Pavel", 6)
pridej_body(t, "Jirka", 2)

body_celkem(t, "Pavel")
body_celkem(t, "Karel")
```
 - Výpis:
Student Pavel má 12 bodů.
Student není v seznamu.