S10 – SD

Alin Zamfiroiu
alin.zamfiroiu@csie.ase.ro

S10 – AVL

- Structura domeniu
- Arbori binari
- Inserare arbore binar
- Numarul de nivele al unui arbore
- Parcurgere arbore binar
- Arbori binari echilibrati
- Moduri de echilibrare

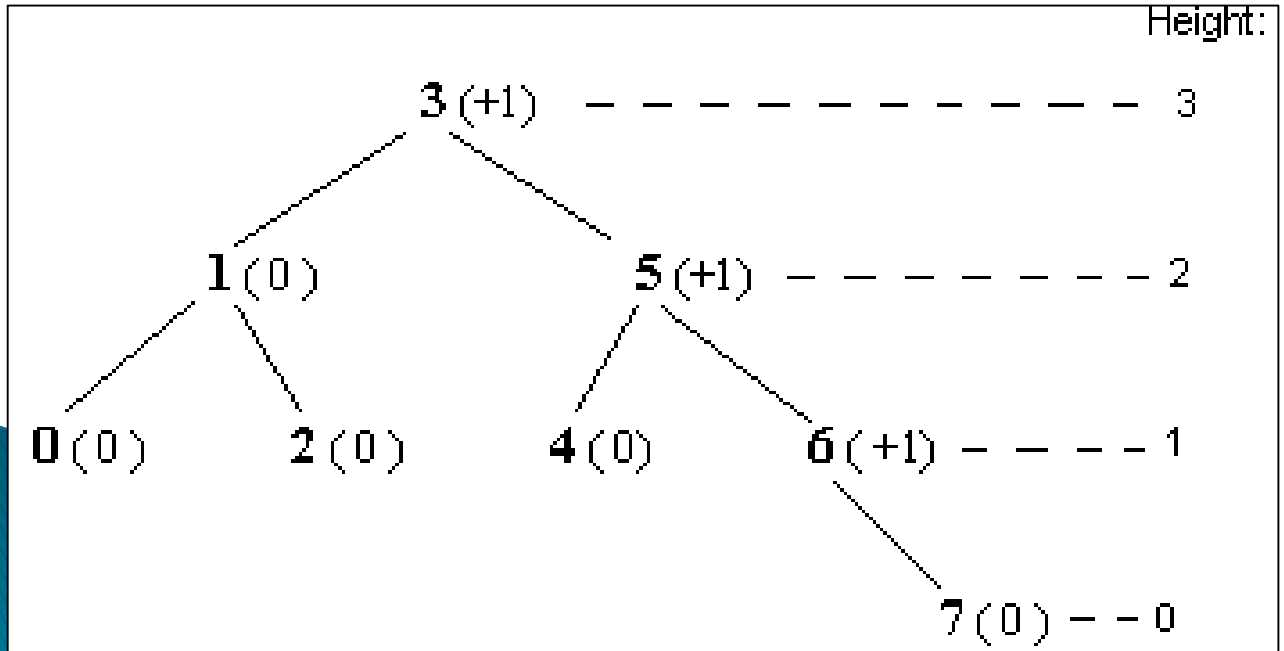
S10 – AVL

```
struct domeniu
{
    int id;
    char* nume_domeniu;
};
```

- Se creeaza si structura pentru un nod al unui arbore binar.
- Se implementeaza metoda de inserare a unui nod intr-un arbore binar de cautare.

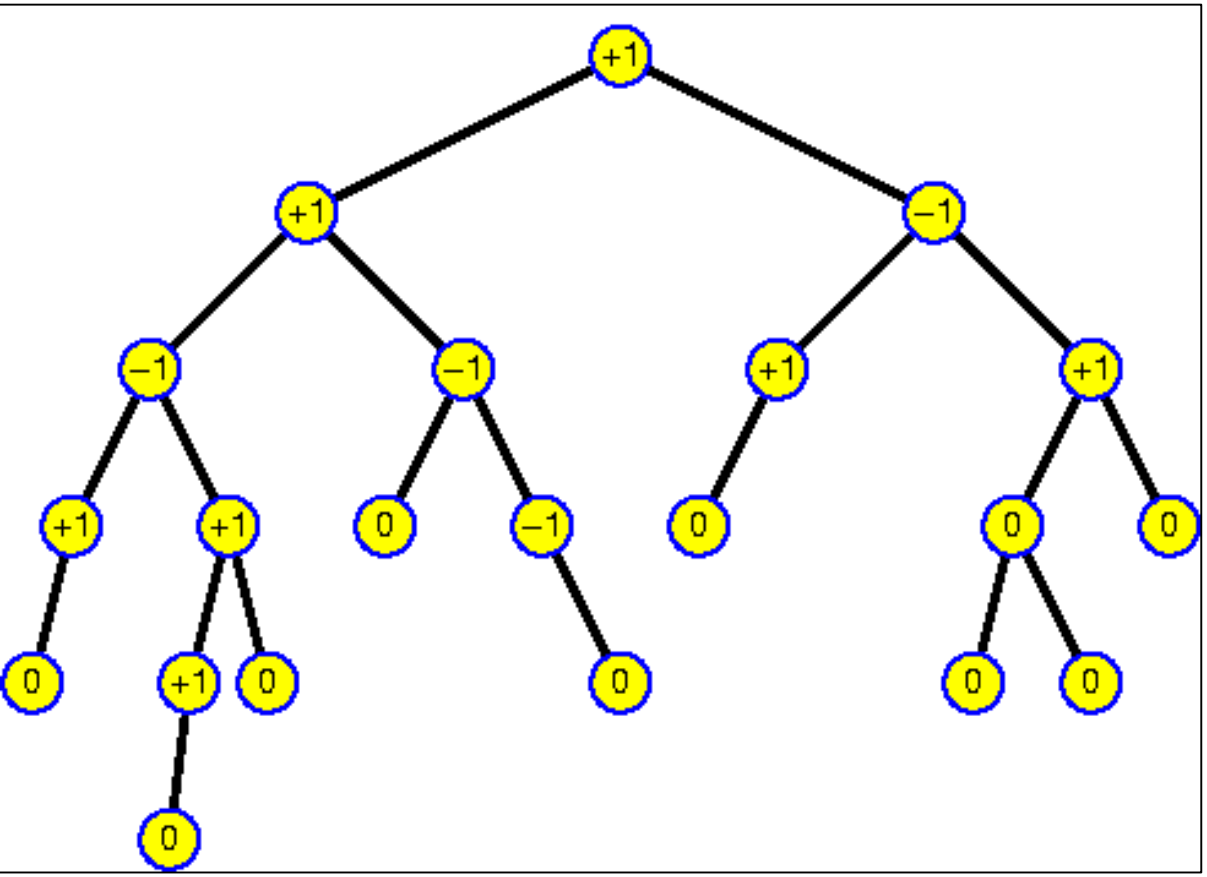
S10 – AVL

- Se implementeaza metoda de determinare a inaltimii unui arbore sau a numarului de nivele
- Se implementeaza metoda de parcurgere in preordine a arborelui binar de cautare.

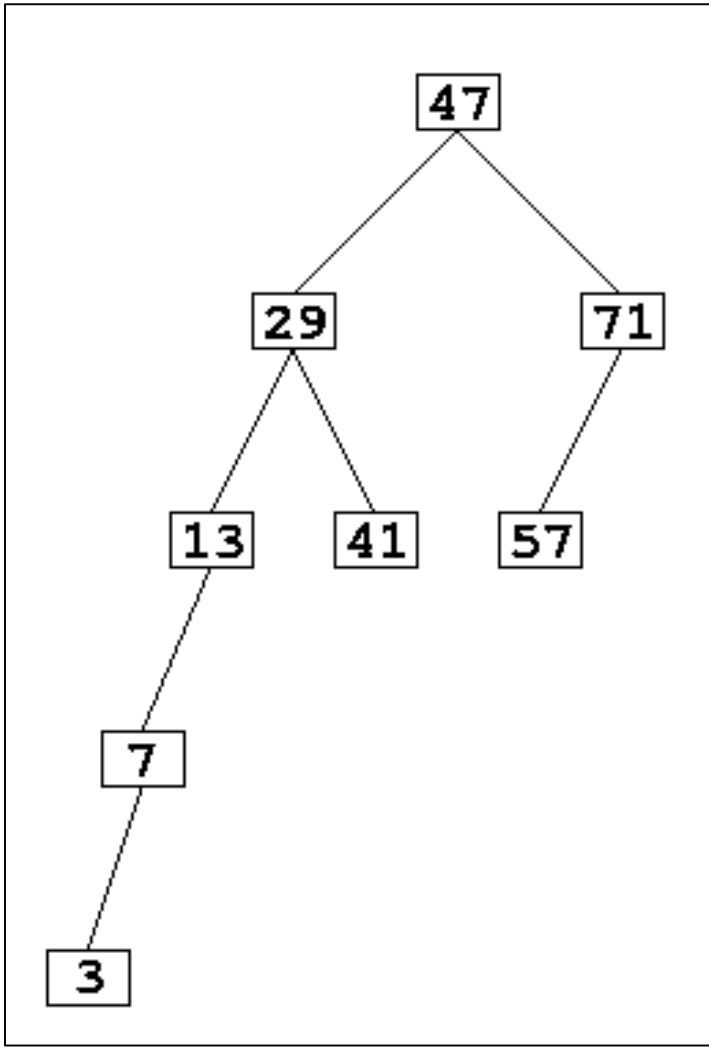


https://www.rtmath.net/

S10 – AVL

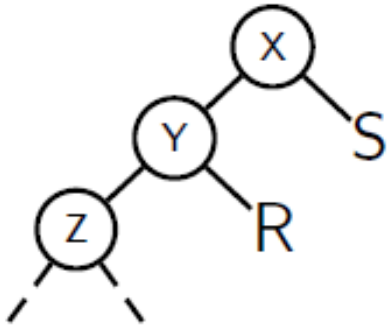


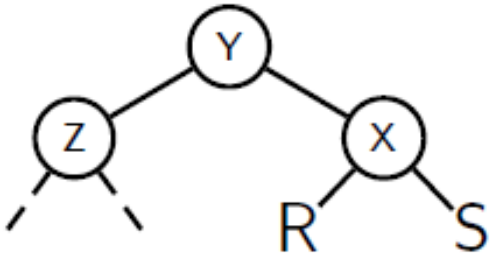
http://users.informatik.uni-halle.de/



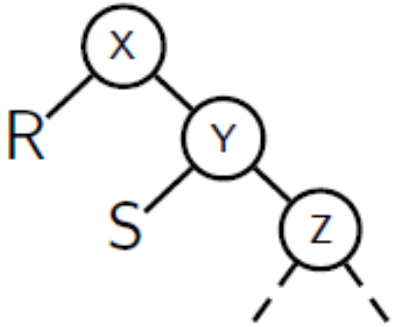
http://pages.cs.wisc.edu/

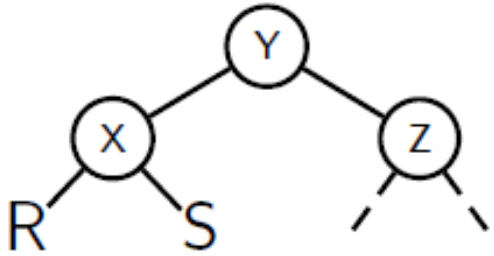
right-rotate(x) →



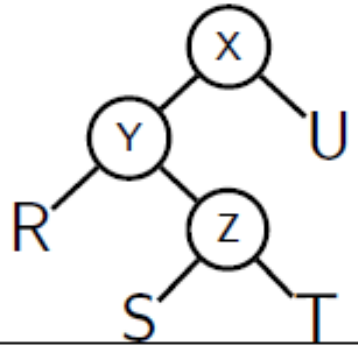


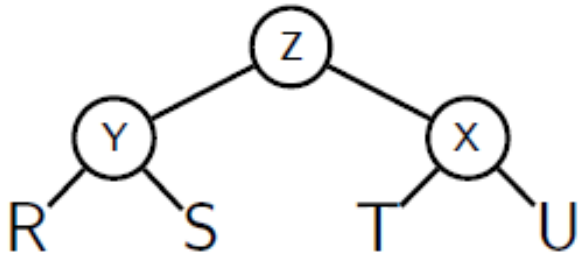
left-rotate(x) →



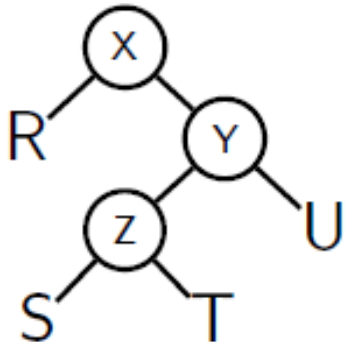


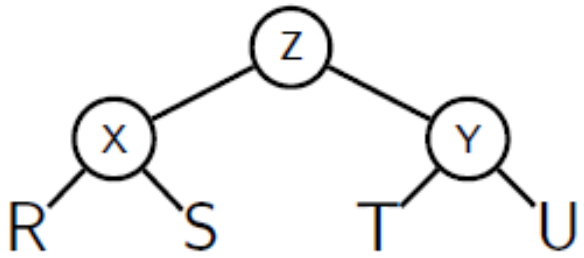
right-double-rotate(x) →





left-double-rotate(x) →





http://www.ambesty.com/

S10 – AVL

```
case 2:
    {//avem arbore dezechilibrat in partea stanga
        nod_arbore * aux;
        aux=rad->st;
        if(factor_balansare(aux)==1)
        {
            //este cazul simplu in care avem doar rotatie dreapta
            rad->st=aux->dr;
            aux->dr=rad;
            rad=aux;
        }
```

S10 – AVL

```
    else
    {
        //este cazul complex in care avem dubla rotatie(stanga-dreapta)
        //stanga
        nod_arbore* aux2=aux->dr;
        aux->dr=aux2->st;
        aux2->st=aux;
        //dreapta
        rad->st=aux2->dr;
        aux2->dr=rad;
        rad=aux2;
    }
}
```