Reproducción en el laboratorio: justificació de les operacions

# La classe Organisme

## L’operació estirar\_organisme

Aquesta operació fa una estirada a l’organisme només si abans no ha estat retallat prèviament. Per fer-ho utilitza la operació “estirar\_recursiu”. En el programa principal es crida aquesta operació mitjançant la instrucció -1.

### Implementació

/\*

Pre: L’organisme ha de tenir una cèl·lula o més

Post: Totes les cèl·lules de l’organisme que no s’han dividit han passat a tenir

dues cèl·lules filles a l’àrbre de cèl·lules de l’organisme

\*/

void Organisme::estirar\_organisme()

{

if (not retallat) estirar\_recursiu(cels, max\_id, tamany);

}

### Justificació

Només es fa una crida a una operació recursiva dins d’aquesta funció en el cas que l’organisme no hagi estat retallat abans. La crida a la funció “estirar\_recursiu” compleix com a condició que l’arbre no estigui buit ja que si l’organisme té una cèl·lula o més, que ve donat per la precondició de la funció, i no ha estat retallat, que ens ho verifica el booleà retallat. A més a més max\_id i tamany són variables inicialitzades ja que la precondició de cridar a la funció estirar\_organisme és que l’organisme tingui una cèl·lula o més i això implica que aquestes variables estaran inicialitzades.

## L’operació estirar\_recursiu

L’operació estirar\_recursiu aplica una estirada a l’organisme fent que totes les cèl·lules que no s’havien dividit abans passin a tenir dues cèl·lules iguals a aquesta però amb un ID superior a la cèl·lula amb l’ID més gran de tot l’organisme. Aquestes dues cèl·lules noves es col·loquen com a filles a l’arbre de cèl·lules de l’organisme.

### Implementació

/\*

Pre: max\_id i tamany estan inicialitzats i no són negatius i 'a' és un

arbre no buit.

Post: Totes les cèl·lules que no s'havien dividit abans han passat a

estar dividides.

\*/

void Organisme::estirar\_recursiu(Arbre<Celula> &a, int &max\_id, int &tam)

{

if (not a.es\_buit())

{

Arbre<Celula> a1, a2;

Celula c = a.arrel();

a.fills(a1, a2);

if (a1.es\_buit() and a2.es\_buit())

{

tam += 2;

Celula aux = c;

Arbre<Celula> buit;

// Plantem els dos arbres i ens estalviem cridar la funció

// recursiva per poder duplicar només la cèl·lula arrel

++max\_id;

aux.id = max\_id;

a1.plantar(aux, buit, buit);

++max\_id;

aux.id = max\_id;

a2.plantar(aux, buit, buit);

}

else

{

estirar\_recursiu(a1, max\_id, tam);

estirar\_recursiu(a2, max\_id, tam);

}

a.plantar(c, a1, a2);

}

}