

Ejercicio Modelo Cálculo Lambda

Preguntas: conteste en JS-ES6 y en estilo FP-OOP

Estudie con la ayuda del profesor `lc.js` (un modelo simple del cálculo lambda en JS) y el caso de uso `test_lc.js`. Implemente los métodos pedidos de forma que el caso de uso funcione. Las salidas que dicen error deben funcionar. Maximice en lo posible el uso de “features” DRY de ES6.

Salida actual

```
PP:node test_lc.js

>>> Testing Lambda Calculus (LC) <<<

Test Case "toString of T"
>>> term.toString( ) = (((\X.(\Y.(X Y))) a) b) <<<

Test Case "T equals T"
>>> term.equals( (((\X.(\Y.(X Y))) a) b) ) = true <<<

Test Case "not equals G"
>>> term.equals( (\X.(\Y.(X Y))) ) = false <<<

Test Case "equals clone"
>>> term.clone( (((\X.(\Y.(X Y))) a) b) ) = (((\X.(\Y.(X Y))) a) b)

Test Case "isClone of TC"
ERROR *** Term.isClone not (correctly) implemented! ***

Test Case "freeVars of T"
ERROR *** Term.freeVars not (correctly) implemented! ***

Test Case "depth of T"
ERROR *** Term.depth not (correctly) implemented! ***

Test Case "replace Y by b in F"
>>> term.replace( {what:X, by:b , except: } ) = (\Y.(b Y)) <<<

Test Case "reduce T Lazy"
>>> term.reduce( {level:10}} ) = (a b) <<<

Test Case "reduce T Eager"
ERROR *** Term.reduce_eager not (correctly) implemented! ***
```