

Auxiliar Nueve - CC41A
Lenguajes de Programación
puzzling with Y?

Oscar E. A. Callaú
oalvarez@dcc.uchile.cl

Santiago - Chile, 20/Oct/2008

1. P1

Con la siguiente definición

```
<RCFAE> ::= .. | {rec {<id> <RCFAE>} <RCFAE>}
```

que expresiones no son válidas, pero aceptables por la definición? Como solucionaría el problema (de una idea de esa implementación)?.

2. P2

Elabore un diagrama que muestre todos los pasos o estados por el cual el ambiente (*Env*) haya pasado con la siguiente expresión.

```
{rec {fact {fun {x} {if0 x 0 {* x {fact {- x 1}}}}} }  
{fact 4}}
```

3. P3

Benancio, proponer hacer el siguiente cambio:

```
(define (cyclically-bind-and-interp bound-id named-expr env)  
  (local ([define value-holder (numV 1729)]  
          ..)  
    (begin  
      (set! value-holder named-expr-val)  
      ..)))
```

responda a las siguientes preguntas, justificando las de ser necesario:

1. Explique que intenta hacer Benancio?
2. Como modificaría su interprete RCFAE, para que funcione adecuadamente según la idea inicial de Benancio?

4. P4

En las clases ud. a podido apreciar el uso de `set!`, del mismo modo en anteriores auxiliares se les ha pedido que implemente

counter , Implementa un conjunto de funciones que permita manejar una estructura contador, ej.:

```
> (define c (new-counter))
> (inc c)
1
> (inc c)
2
> (dec c)
1
...
```

Stack , Usando la misma técnica anterior implemente una pila con las funciones::

(new-stack) , retorna una pila vacía.

(push stack val) , agrega `val` a la pila `stack`.

(pop stack) , saca el elemento de encima de la pila y lo retorna.

(peek stack) , retorna el valor encima de la pila `stack`, sin modificarla.

Haga estas implementaciones de nuevo, sin usar `box`, usando `set!`.