

1- (3 pontos)

Compare with com rec. São essencialmente a mesma coisa no nível sintático.

- qual a diferença e por quê precisamos do rec?
- seria possível suprimir o rec tal fizemos com o with?
- em quais situações podemos trocar with por rec sem afetar o resultado?

2- (2 pontos)

A função cyclically-bind-and-interp usa um truque para evitar o problema da auto-referencia. Explique o funcionamento deste truque.

3- (1 ponto)

É verdade que a ordem de avaliação só é importante se tivermos armazenagem (store) além de ambiente (environment)? Por quê ?

4- (1 ponto)

Faz sentido em limpar o environment, tal como liberamos o store na coleta de lixo? Em quais condições

5- Vimos que para "criar" uma função recursiva a técnica consiste em construir uma função que é aplicada a si mesma:

```
(lambda (p)
  ((lambda (f) (f f))
   (lambda (f) (p (f f)))))
)
```

No entanto, o combinador Y tem a seguinte forma, por quê?

```
(lambda (p)
  ((lambda (f) (f f))
   (lambda (f) (p (lambda (a) ((f f) a))))))
)
```

6- (2 pontos)

Qual deve ser o tipo de X para que a seguinte expressão seja coerente? demonstre com a avaliação de tipos.

```
{fun {x:tipo}: number}{+ {x x}3}}
```