

Universidad Nacional Autónoma de México

Facultad de Ciencias

Lenguajes de Programación

Tarea 3

PROBLEMA I

$$\frac{\text{FIBONACCI} \quad \Gamma}{\Gamma I - (\text{rec}(f \text{ ibo} : \text{number}))}$$

$$\frac{\text{EMPTY} \quad \Gamma l - l : \text{List}}{\Gamma l - (\text{Empty} ? l) : \text{Bool}}$$

PROBLEMA II

(1) (+ (2)1 (3)(first (4)(cons (5>true (6)empty)))
[[(1)]] = (+ (2) (3))
[[(2)]] = [[1]] → number
[[(3)]] = (first (4) (cons (5>true (6)empty))) → number
[[(4)]] = (cons (5>true (6)empty)) → nlist
[[(5)]] = [[true]] → bool
1[[(6)]] = [[empty]] → nlist

En [[(5)]] vemos que se está intentando construir una lista de números con un tipo bool lo cual causaría un error

PROBLEMA III

[1] = [F] → number
[2] = [x] → number
[3] = [y] → number
[4] = [cos x {f {f y}}] → list ya que [x] → number y {f {f y}} → list
[5] = [x] → number
[6] = [{f {f y}}] → list
[7] = [{f y}] → nlist

Por lo tanto C_1 , C_3 , C_5 son tipo number y C_2 , C_4 , C_6 son nlist.

PROBLEMA IV

No importa si es perezoso o glotón Los juicios de tipo no cambian ya que el chequeo de tipos no se realiza en tiempo de ejecución, lo hace el compilador o interprete, entonces la evaluación no afecta el chequeo de tipos.

PROBLEMA V

Polimorfismo Explícito

Ventajas:

- Es más rápido en tiempo de compilación.
- Podemos usar el mismo código para varios propósitos.

Desventajas

- Escribir el código fuente es más tardado.
- una variable no puede ser utilizada para distintos tipos lo que significar más variables y por ende más espacio en memoria ocupado.

Polimorfismo Implícito

Ventajas

- No hay necesidad de construir un tipo en específico ya que siempre que usemos variables estaremos trabajando con tipos genéricos.
- El código fuente es mucho menos verboso.

Desventajas

- Los errores de tipo se dan en tiempo de ejecución.
- El programa es más propenso a errores de semántica.

PROBLEMA VI

DSL

Ventajas

- Fácil para resolver problemas específicos ya que es más "liviano" que un lenguaje de propósito general.
- Resuelven el problema sin redundancias.

Desventajas:

- Los problemas más grandes requieren el uso de un lenguaje de propósito general

Lenguajes de propósito general

Ventajas

- DSL dentro del mismo lenguaje de propósito general.
- Mayor control sobre el programa.

Desventajas:

- A veces hay muchas cosas innecesarias.
- A veces es menos eficaz para resolver problemas pequeños y específicos.