

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)))
 $\llbracket 1 \rrbracket = (+ (2) (3))$
 $\llbracket 2 \rrbracket = \llbracket 1 \rrbracket \rightarrow \text{number}$
 $\llbracket 3 \rrbracket = (\text{first} (4) (\text{cons} (5) \text{true} (6) \text{empty})) \rightarrow \text{number}$
 $\llbracket 4 \rrbracket = (\text{cons} (5) \text{true} (6) \text{empty}) \rightarrow \text{nlist}$
 $\llbracket 5 \rrbracket = \llbracket \text{true} \rrbracket \rightarrow \text{bool}$
 $\llbracket 6 \rrbracket = \llbracket \text{empty} \rrbracket \rightarrow \text{nlist}$

En $\llbracket 5 \rrbracket$ vemos que se está intentando construir una lista de números con un tipo bool lo cual causaría un error

PROBLEMA III

$\llbracket 1 \rrbracket = [F] \rightarrow \text{number}$
 $\llbracket 2 \rrbracket = [x] \rightarrow \text{number}$
 $\llbracket 3 \rrbracket = [y] \rightarrow \text{number}$
 $\llbracket 4 \rrbracket = [\text{cons } x \{f \{f y\}\}] \rightarrow \text{list}$ ya que $[x] \rightarrow \text{number}$ y $[\{f \{f y\}\}] \rightarrow \text{list}$
 $\llbracket 5 \rrbracket = [x] \rightarrow \text{number}$
 $\llbracket 6 \rrbracket = [\{f \{f y\}\}] \rightarrow \text{list}$
 $\llbracket 7 \rrbracket = [\{f y\}] \rightarrow \text{nlist}$

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 significa 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.