Γ ← ⊢

Lenguajes de Programación
Tarea 3

Bravo García Marco Antonio
Herrera Brito Juan Jose

Problema I

Fibonacci

Γ ⊢ fib: (number ← number) Γ ⊢ n:number

───

Γ ⊢ [fib ← number]   Γ ⊢ {fib n}:number   [A]

──

Γ ⊢ {rec {fib: number
                {fun (n:number):number
                        {if (<= n 1)
                                1
                        (+ (fib (- n 1)) (fib (- n 2)))}}}}{fib n}

                                                              Γ ⊢ n:number
        Γ ⊢ n:number   Γ ⊢ 1:number   Γ ⊢ n:number Γ ⊢ 1:number   Γ ⊢ 2:number

        ─────────────────────────   ──────────────────────   ────────────────────

        Γ ⊢ (<= n 1):bool   Γ ⊢ (fib (- n 1))   Γ ⊢ (fib (- n 2))

        ──────────────────   ──────────────────────────────────────

        Γ ⊢ (<= n 1):bool   Γ ⊢ 1:number   Γ ⊢ (+ (fib (- n 1)) (fib (- n 2)))

        ──

        Γ, [fib ← number] ⊢ (if (<= n 1)
                                        1
                        (+ (fib (- n 1)) (fib (- n 2))))

        ──

[A]= Γ, [fib ← number] ⊢ (fun (n:number):number : number → number
                                {if (<= n 1)
                                        1
                                (+ (fib (- n 1)) (fib (- n 2)))})

Empty List

Γ ⊢ empty?:(list → bool)   Γ ⊢ l:list

──────────────────────────────────────

Γ ⊢ (empty? l)

Problema II

❶(+ ❷1 ❸(first ❹(cons ❺ true ❻empty)))

[❶] = (+ 1 (first (cons true empty)))

[❶] = number y [❷] = [❸] = number

[❷] = [1] = number

[❸] = [❹] → [❺]

[❹] = list y [❺] = number [❻] = list

[❺] = [true] = bool

Aqui hay una inconsistencia en los tipos y da un error

Problema III
Problema IV
Problema V
Problema VI