Lenguajes de programación

Recursividad Profunda

Listas en Scheme

Tipos de problemas para manejar listas

- Problemas que trabajan con **listas imbricadas (listas que contienen listas)** y se desea llegar hasta los átomos: **RECURSIVIDAD PROFUNDA**
  - Problemas que reciben al menos una lista imbricada y generan un resultado atómico.
  - Problemas que reciben valores atómicos y generan una lista imbricada.
  - Problemas que reciben al menos una lista imbricada y generan una lista.

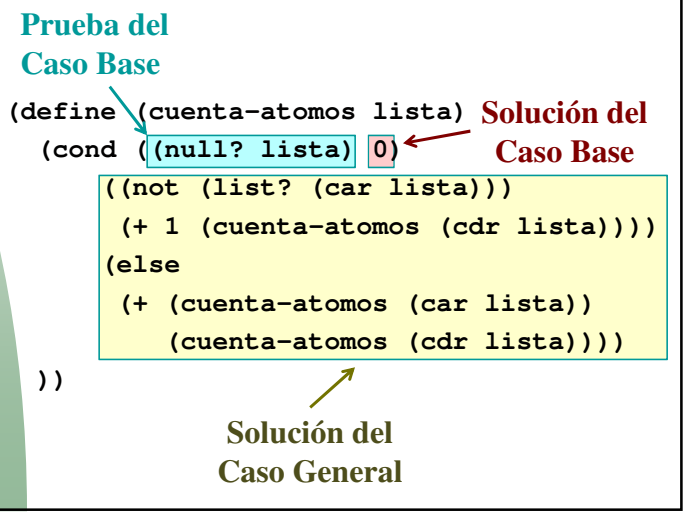
RECURSIVIDAD PROFUNDA

- Igual que en los casos de recursividad plana, pero validando si el elemento a analizar es atómico o no.
- Si es atómico se utiliza para obtener la solución buscada.
- Si no es atómico, se aplica la recursividad sobre ese elemento, y sobre el resto de la lista para obtener la solución general

Ejemplo: Lista → átomo

- Contar los elementos atómicos de una lista imbricada
  (cuenta-atomos '(1 2 3 4)) → 4
  (cuenta-atomos '(a (b c (d)))) → 4
  (cuenta-atomos '()) → 0
- **CASO BASE:** lista vacía
  # elementos = 0
- **CASO GENERAL**: lista con elementos
  Si el primer elemento es atómico
  # elementos = 1 (primer elemento) +
  cantidad de átomos en el resto de la lista
  Si no
  # elementos =
  cantidad de átomos en el primer elemento +
  cantidad de átomos en el resto de la lista

Solución de Ejemplo



Ejemplo: Átomo→ Lista

- Generar una lista donde su único elemento "n" se encuentre anidado "n" veces.
  (anida-nveces 0) → (0)
  (anida-nveces 2) → (((2)))

- Llamar a la función recursiva auxiliar:
  (anida-aux n veces-anidada) , iniciando en 0 veces anidada
- **CASO BASE**: n = veces-anidada
  Resultado = (n)
- **CASO GENERAL**: n > veces-anidada
  Resultado = anidar lista con n anidada n-1 veces

Solución de Ejemplo

```
(define (anida-nveces n)
  (anida-aux n 0))

(define (anida-aux n veces-anidada)
  (if (= n veces-anidada)
      (list n)
      (list (anida-aux n
                       (+ veces-anidada 1)))
  ))
```

Prueba del Caso Base

Solución del Caso Base

Solución del Caso General

Ejemplo: Lista→ Lista

- Incrementar los elementos de una lista imbricada
  (incrementa-li '()) → ( )
  (incrementa-li '(1 (2 (3) 4) 5) → (2 (3 (4) 5) 6)

- CASO BASE = lista vacía
  Resultado = lista vacía = ()
- CASO GENERAL = lista con elementos
  Si el primer elemento es un número
  Resultado = agregar el primer elemento incrementado a una lista con el resto de los elementos con sus números incrementados
  Si no
  Resultado = agregar los elementos numéricos del primer elemento incrementado a una lista con el resto de los elementos con sus números incrementados

Solución de Ejemplo