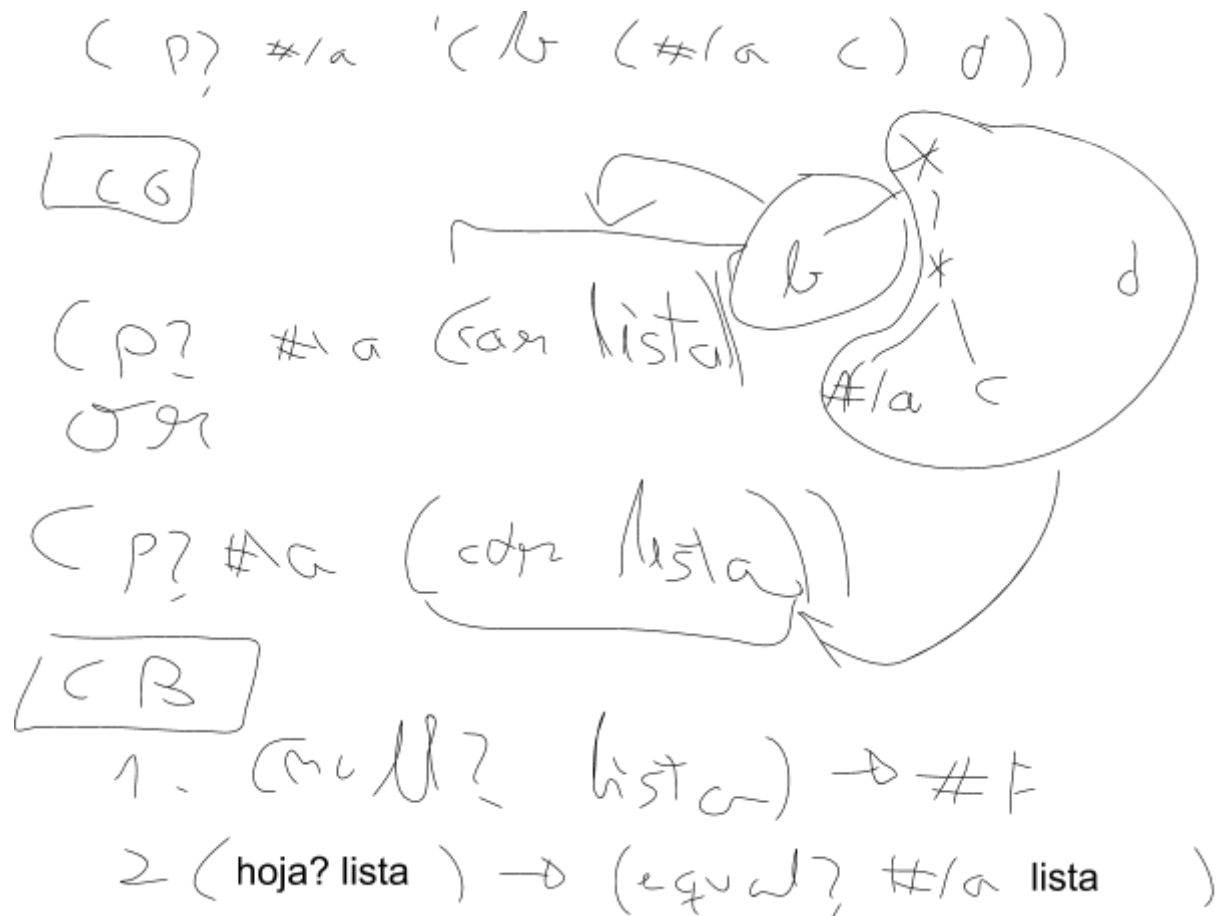


Funciones sobre listas estructuradas

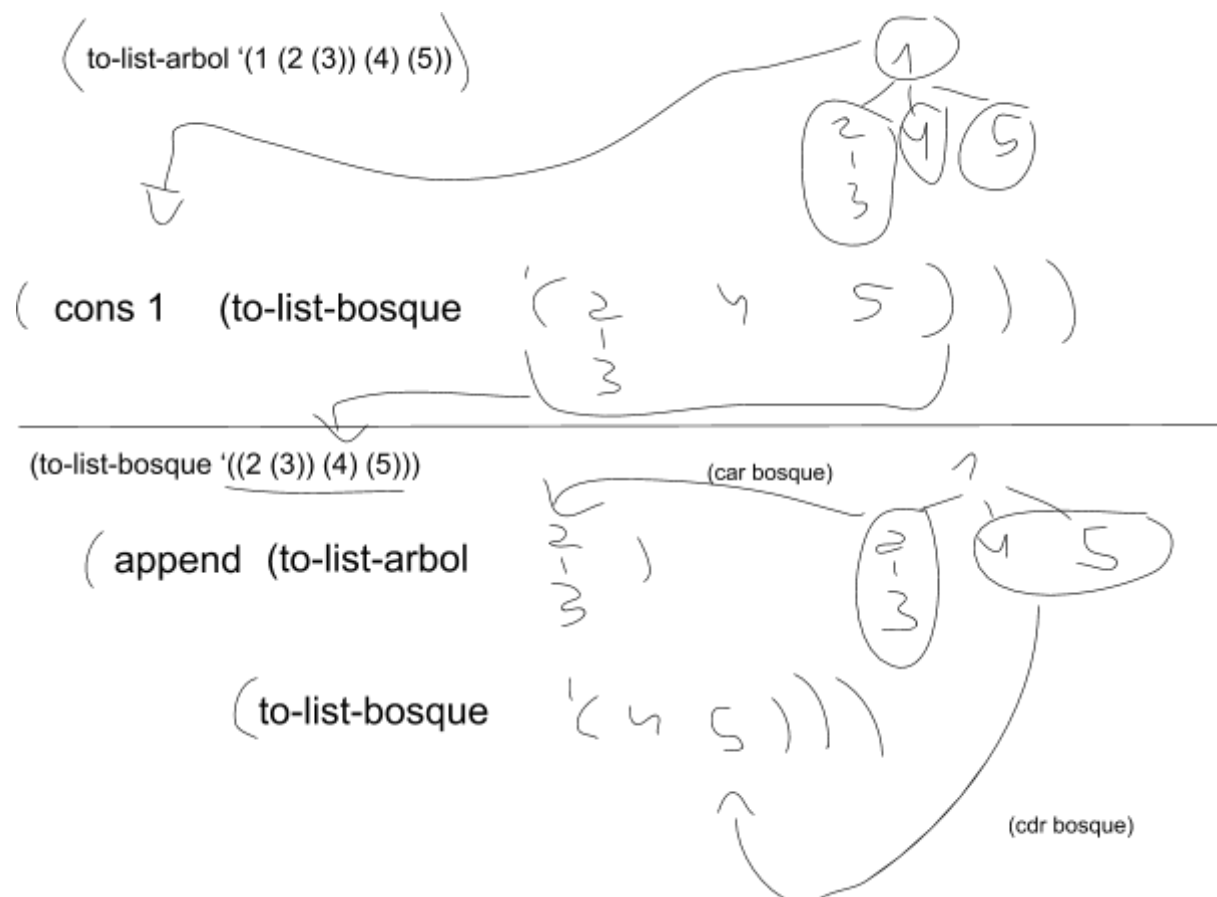
Ejercicio: Diseña la función (pertenece? dato lista) que comprueba que un dato se encuentra en una lista.



```
(define (pertenece? dato lista)
  (cond ((null? lista) #f)
        ((hoja? lista) (equal? dato lista))
        (else (or (pertenece? dato (car lista)) (pertenece? dato (cdr lista))))))
```

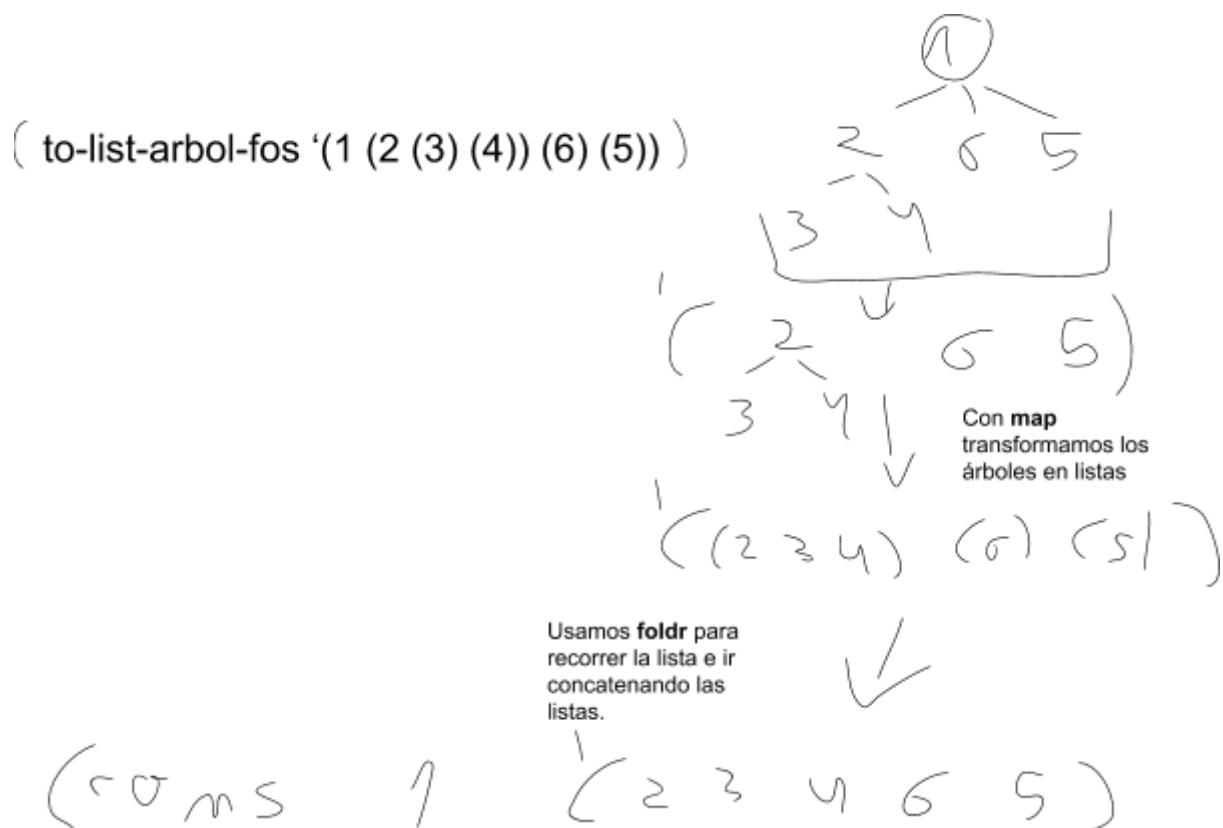

Funciones sobre árboles

Ejercicio: Diseña la función (to-list-arbol arbol) que devuelve una lista con los elementos del árbol.



```
(define (to-list-arbol arbol)
  (cons (dato-arbol arbol)
        (to-list-bosque (hijos-arbol arbol))))
(define (to-list-bosque bosque)
  (if (null? bosque)
      '()
      (append (to-list-arbol (car bosque))
                (to-list-bosque (cdr bosque)))))
```

Ejercicio: Diseña la función (to-list-arbol-fos arbol) que devuelve una lista con los elementos del árbol usando FOS.



```
(define (to-list-arbol-fos arbol)
  (cons (dato-arbol arbol)
        (foldr append '()
                (map to-list-arbol-fos (hijos-arbol arbol)))))
```

Bibliografía

Los enunciados de los ejercicios resueltos, y los resúmenes, se han elaborado a partir del material publicado en <https://domingogallardo.github.io/> , material del que es propietario el Departamento de Ciencia de la Computación e Inteligencia Artificial de la Universidad de Alicante, Domingo Gallardo, Cristina Pomares, Antonio Botía y Francisco Martínez.