Lenguajes de Programación
Auxiliar No1

Auxiliares: Kenji Maillard

13/03/2020

1 Resumen

- Tipos Primitivos: Números, booleanos, Strings y símbolos
- Funciones predefinidas: +, -, equal?, substring, ...
- Condicionales:
  - ```
    (if a
        TRUE_BRANCH
        FALSE_BRANCH)
    ```
  - ```
    (cond  [(> 2 3)  (printf "hola")]
           [(> 6 5)  (printf "chao")]
           [else #f])
    ```
- Definir identificadores: `(define a 2)`
- Definir funciones::

  ```
  (define (double x)
    (+ x x))
  ```
- Estructuras de datos:
  - Pares: `(cons 1 2)` o `'(1 . 2)`
  - Listas: `(list 1 2 3)` o `'(1 2 3)`

Si tiene una duda sobre una función, puedes usar el Help Desk (menú Help en DrRacket).
Recuerde enunciar el contrato y escribir tests.

2 Ejercicios

1. Conceptos:
   (a) ¿Cual es la diferencia entre `(cons 'a 'b)` y `(list 'a 'b)`? ¿Como se representaria el segundo con notación de pares?
   (b) ¿Cual es la notación de lista equivalente a `'((a b) c)` ?
   (c) Dado `(define l (list '(a b c) '(d e f) '(g h i)))`, ¿Como se accesaria el elemento `'c` y el `'e` en l? Por ejemplo, `'b` es accesado por `(car (cdr (car l)))`
   (d) Usando solo `cons`, la lista vacia y simbolos, muestre como construir las siguientes expresiones: `'(c)`, `'(a b)`, `'((a b) (c))`

2. Defina la función `pair-add1 p` que recibe un par de números y retorna un nuevo par dónde los dos elementos fueron incrementados en `1`.

3. Defina las siguientes funcionalidades utilizando `map`, `foldl`, or `filter`:
   (a) Dado una lista de enteros, retorne una lista con los elementos incrementados en uno.
   (b) Dado una lista de strings, retorne una lista con la longitud de cada cadena.
   (c) Dado una lista de enteros, retorne la suma de todos su elementos.
   (d) Dado una lista de string, retorne el string resultante de la concatenacion de todas las cadenas de la lista.
   (e) Dado un lista de enteros, retorne la lista de todos los elementos mayores que cero.

4. Defina las funciones `map`, `foldl` y `foldr` sobre listas.

5. Un árbol binario se puede describir con la BNF sigiente:
   $\langle bt \rangle$ ::= $\langle val \rangle$
   | `(` `cons` $\langle bt \rangle$ $\langle bt \rangle$ `)`
   (a) Defina las funciones `leaf`, `node`, `node?` para construir y analyzar un árbol. ¿Porqué la función `leaf?` es inutil?
   (b) Siguiendo los ejemplos para naturales y listas, describe el patrón de una función recursiva sobre un árbol binario.
   (c) Utilize el patrón para definir una funcíon `bt-contains?` que determina si un árbol contiene un elemento.

6. Funciones sobre árboles binarios.
   (a) Defina las funciones `map`, `fold` sobre árboles binarios. Para `fold`, pense bien en lo que necesite para agregar los datos.
   (b) Utilizando `fold`, defina las funciones `exists` y `forall`.
   - `exist` verifica si un elemento satisfece un predicado dado.
   - `forall` verifica si todos los elementos satisfecen el predicado.