

Estructuras Discretas 2023-1

Práctica 3: Listas de longitud par.

Lourdes del Carmen González Huesca
María Fernanda Mendoza Castillo

Juan Alfonso Garduño Solís
Alan Moreno de la Rosa

26 de septiembre de 2022

Fecha de entrega: Sábado 8 de octubre de 2022

Introducción.

Las listas de longitud par, tienen las mismas características que las listas que ya hemos revisado en el curso, la única peculiaridad como su nombre lo indica, es que la cantidad de elementos dentro de la lista debe ser par.

Para cumplir con este requisito, se debe cuidar que todas las operaciones sobre estas listas preserven el invariante de que la longitud sea par. Por ejemplo, no es posible definir una función que agregue un único elemento a la lista.

Implementación.

Como se mencionó, podemos añadir elementos a la lista, pero para conservar su propiedad de longitud par, siempre debemos añadir 2 elementos. Para facilitar esta acción veremos esos elementos como una unidad, añadiendo un par (a,b), utilizando los pares que Haskell nos ofrece. Los elementos podrán buscarse de manera individual, es decir, a pesar de que la implementación sea con pares, conceptualmente se pensará en una lista de elementos consecutivos.

Es decir, la lista [1,2,3,4,5,6] estará dada por [(1,2),(3,4),(5,6)].

Así que utilizaremos el siguiente alias:

```
type EList a = [(a,a)]
```

Ejercicios:

Define las siguientes funciones definiendo la **definición de listas**.

longP Dada una EList obtiene su número de elementos.

```
longP [(1,2),(3,4),(5,6)] = 6
```

elemP Dado elemento e y una EList l, verifica si e pertenece a l.

```
elemP 3 [(1,2),(3,4),(5,6)] = True
```

```
elemP 9 [(1,2),(3,4),(5,6)] = False
```

consP Dados dos elementos y una EList, los agrega al principio de la EList.

```
consP (1,2) [(7,8),(9,10),(11,12)] = [(1,2),(7,8),(9,10),(11,12)]
```

```
consP (232,23) [] = [(232,23)]
```

appendP Función que realiza la concatenación de dos EList.

```
appendP [(1,2),(3,4),(5,6)] [(7,8),(9,10)] = [(1,2),(3,4),(5,6),(7,8),(9,10)]
```

snocP Función que agrega dos elementos al final de una EList.

```

snocP (2,1) [(6,5),(4,3)] = [(6,5),(4,3),(2,1)]
snocP (12,754) [(7,8),(9,10),(11,12)] = [(7,8),(9,10),(11,12),(12,754)]

atP Dada una EList l y un entero n, regresa el n-ésimo elemento de l.

atP [(7,8),(9,10),(11,12)] 3 = 9

updateP Dada una EList l, un número n y un elemento e, cambia el n-ésimo elemento de l por e.

updateP [(7,8),(9,10),(11,12)] 4 6 = [(7,8),(9,6),(11,12)]
updateP [('h','o'),('l','a')] 3 'y' = [('h','o'),('y','a')]

aplanaP Función que recibe una EList l y regresa la misma pero aplanada, es decir, todos los elementos de l separados en una lista común.

aplanaP [(1,2),(3,4),(5,6)] = [1,2,3,4,5,6]

toEL Función que convierte una lista común en una EList, si la lista común es de longitud impar, no se agregará el último elemento.

toEL [1,2,3,4,5] = [(1,2),(3,4)]
toEL "hola" = [('h','o'),('l','a')]

dropP Dado un entero n y una EList l, borra los n primeros elementos de l, donde n es par.

dropP 2 [('h','o'),('l','a')] = [('l','a')]
dropP 3 (toEL "hola") = ***Exception: No es un numero par

dropN Al igual que el inciso anterior, borra los n primeros elementos de l, pero n puede ser par o impar, pues se regresará una lista común.

dropN 3 [(1,2),(7,8),(9,10),(11,12),(12,754)] = [8,9,10,11,12,12,754]

takeP Dado un entero n y una EList l, toma los n primeros elementos l, n debe ser par.

takeP 2 [(1,2),(3,4)] = [(1,2)]

takeN Al igual que el inciso anterior, toma los primeros n elementos de una EList, pero n puede ser par o impar, pues los elementos se regresarán en una lista común.

takeN 3 [(1,2),(3,4)] = [1,2,3]

reversaP Elabora una función que haga la reversa de una EList.

reversaP [(1,2),(3,4),(5,6),(7,8)] = [(8,7),(6,5),(4,3),(2,1)]

```

Un punto extra: Da la función `map` para esta estructura.

Notas.

- Las instrucciones para entregar las prácticas están en la página del curso, si aún no sabes cuál es [haz click aquí](#).
- Se responden dudas en el correo pero de preferencia enviame mensaje por Telegram. Puedes acceder a mi contacto [haciendo click aquí](#).

No dudes en contactarme ante cualquier duda, aunque no sea de la materia.