

Teoría de Lenguajes

Teoría de la Programación

Clase 3: Técnicas de programación declarativa

Problema 1

Desarrollar un procedimiento que dadas dos listas ordenadas en forma ascendente las mezcle en forma ordenada

Problema 2

Desarrollar un procedimiento que dada una lista la subdivida en 2 listas.

Problema 3

Con lo desarrollado anteriormente, hacer un mergesort.

Tarea para el hogar: Pensar complejidad espacial y temporal del Mergesort

Pensar recursivamente

Una lista es una estructura de datos recursiva. Una función que calcula sobre listas tendrá:

- Caso base
- Caso recursivo

Convertir recursividad a computaciones iterativas! => Tail recursion

- Accumulators!!

High order programming

High order programming

Conjunto de técnicas de programación que se pueden utilizar cuando los procedimientos pueden ser tomados como valores

Un procedimiento cuyos parámetros no son procedimientos, se considera de primer orden

First class citizen

Un “ciudadano de primera clase” es una entidad que soporta todas las operaciones que soporta cualquier tipo básico

En el modelo declarativo, los procedimientos, las funciones, son ciudadanos de primera clase

¿Qué podemos hacer con
high order?

Operaciones básicas

- Abstracción procedural
- Genericidad
- Instanciación
- Embedding

Abstracción procedural

Cualquier statement puede ser convertido a un procedure value

- Statements diferidos
- Parámetros

Genericidad

Enviar procedimientos como parámetro
permite realizar procedimientos genéricos.

1. Sumar todos los elementos de una lista
2. Multiplicar todos los elementos de una lista
3. ¿Cómo lo hacemos genérico?

Instanciación

Devolver un procedure value como resultado de una expresión.

1. Revisar el MergeSort. ¿Cómo lo hacemos genérico?
2. Hacer un factory de esos sorts

Embedding

Poder utilizar procedure values como parte de una estructura de datos

Ej: Lo que vimos como clases en el capítulo 1

Funciones anónimas

Definición de una función sin necesidad de usar un identificador de variable

A veces llamadas funciones lambda

Currying

Técnica mediante la cual todos los procedimientos reciben solo un parámetro.

Si tienen más de un parámetro, en realidad es tomado como una función que tiene a su vez un solo parámetro

Currying

Algunos lenguajes soportan curry en forma natural

Permite evaluar funciones en forma parcial



Haskell Curry (1900 -1982)

Loops

Nueva abstracción lingüística: Declarative loops

- Integer loop : {For 1 10 1 Browse}
- List loop: {ForAll [a b c] Browse}

Imperative vs Declarative loops

Cada iteración declara una nueva variable

Cada iteración es independiente de la otra => Podrían ejecutarse en forma concurrente

Modularizando en Oz

Módulos e interfaces en Oz

Módulo: Colección de procedimientos y valores

Functor: Expresión que especifica los componentes de un módulo

Functors de Oz != Concepto de functors de i.e. Haskell

Bibliografía

- **Concepts, Techniques, and Models of Computer Programming - Capítulo 3**, Peter Van Roy and Seif Haridi
- **Extras:**
 - **Módulos e interfaces de Oz**

<https://mozart.github.io/mozart-v1/doc-1.4.0/tutorial/node7.html>

- **High order & Currying con Haskell**

<http://learnyouahaskell.com/higher-order-functions>