



Evaluación Perezosa

Concepto

Es una estrategia de evaluación que retrasa el cálculo de una expresión hasta que su valor sea necesario, y que también evita repetir la evaluación en caso de ser necesaria en posteriores ocasiones. Esta competición del cálculo puede reducir el tiempo de ejecución de ciertas funciones de forma exponencial, comparado con otros tipos de evaluación.

Ventajas y Desventajas

Los lenguajes perezosos usan evaluación perezosa, es decir, al evaluar una expresión evalúan sus argumentos sólo cuando los necesita. De manera opuesta, en la evaluación impaciente los argumentos de las expresiones se evalúan antes que las expresiones. En esta entrada presento un ejercicio para Informática (del Grado de Matemáticas) con objeto de resaltar la ventaja de la evaluación perezosa de Haskell frente a la evaluación impaciente de Maxima.

Ejemplo

```
def calculateData: String = {
  print("Calculating expensive data!")
  "some expensive data"
}

def dumbMediator(preconditions: Boolean, data:
String): Option[String] = {
  print("Applying mediator")
  preconditions match {
    case true => Some(data)
    case false => None
  }
}

def smartMediator(preconditions: Boolean,
data: => String): Option[String] = {
  print("Applying mediator")
  preconditions match {
    case true => Some(data)
    case false => None
  }
}

smartMediator(preconditions = false,
calculateData)

dumbMediator(preconditions = false,
calculateData)
```

Funciones y Patrones

Funciones de alto orden

Las funciones de alto orden, es un concepto simple pero muy poderoso el cual tienen todos los lenguaje de programación funcionales, es básicamente tratar a las funciones como ciudadanos de primera clase, esto significa ue que las funciones son objetos.

Monada

Una mónada es un patrón de diseño funcional que tiene su origen en la teoría de las categorías, aunque no hace falta ser ningún experto en matemáticas para utilizarlas. Una mónada siempre encapsula un tipo de datos que nosotros elegimos para crear instancias de otro tipo nuevo asociado a una computación especial. Es común que dicha computación maneje un caso especial de dicho tipo. Una mónada siempre define un tipo de datos y cómo podemos combinar los valores de dicho tipo.

Monoides

Los monoides son sistemas algebraicos compuestos por (A, 0) el que cumplen con la estructura de Semigrupo y añaden una nueva restricción: el elemento neutro.

Referencias

Jose Luis Rodriguez. (2018). Álgebras y funciones: patrones en programación funcional. 6 de marzo del 2020, de adictos al trabajo Sitio web: <https://www.adictosaltrabajo.com/2018/08/14/algebras-y-funciones-patrones-en-programacion-funcional/>

Yeison Daza. (2016). Funciones de alto orden en JavaScript. 6 de marzo del 2020, de medium Sitio web: <https://medium.com/entendiendo-javascript/funciones-de-alto-orden-en-javascript-42d04769d9b5>

Jorge Aguado Praena. (2018). ¿Qué es una mónada?. 6 de marzo del 2020, de Thoughts on Software Sitio web: <http://joragupra.com/2018/01/que-es-una-monada.html>

Daniel Verela. (2015). Programación Funcional con Scala. 6 de marzo del 2020, de Nearsoft Jobs Sitio web: <https://blog.nearsoftjobs.com/programaci%C3%B3n-funcional-con-scala-4708b6b56e30>