Meeting 16 - Higher-Order Fun

Bor-Yuh Evan Chang

Saturday, October 12, 2024

Meeting 16 - Higher-Order Fun

What questions does your neighbor have?

Links

In-Class Slides

In-Class Jupyter

Book Chapter

Announcements

- HW4 due Mon 10/21 6pm
- Exam 1-2 feedback released.
  - See CM Lawrence about errors grading.
  - See me and TAs about how to improve for next time.
- Prof. Chang traveling next week

Today

- Higher-Order Functions (mini-lecture)
- Triage Your Questions
  - HW4?

Questions?

- Review:
  - What is a *higher-order function*?

Questions?

Higher-Order Functions

Currying

```
1  0
```

```
res0: Int = 0
```

Collections and Callbacks

Map

```
1  0
```

```
res7: Int = 0
```

FlatMap

```scala
def map[A, B](l: List[A])(f: A => B): List[B] = l match {
  case Nil => Nil
  case h :: t => f(h) :: map(t)(f)
}
```

```
defined function map
```

FoldRight

FoldLeft

Abstract Data Types

```
1  val m = Map(2 -> List("two", "dos", "二"), 10 -> List("ten", "diez", "十"))
```

```
m: Map[Int, List[String]] = Map(
  2 -> List("two", "dos", "\u4e8c"),
  10 -> List("ten", "diez", "\u5341")
)
```

Parallel and Distributed

Scala Parallel Collections Library

Run the following cell to load the Scala Parallel Collections library.

```
scala.collection.parallel.CollectionConverters._
```

```
1  import $ivy.`org.scala-lang.modules::scala-parallel-collections:1.0.4`, scala.collection.parallel.Collect
```

```
import $ivy.$                                                    , scala.collection.parallel.CollectionCo
```

```
1  val par0to9999 = (0 to 9999).toList.par
2  val sum = par0to9999.map(_ + 1).reduce(_ + _)
3  assert(sum == 50005000)
```

```
par0to9999: collection.parallel.immutable.ParSeq[Int] = ParVector(0, 1, 2, 3, 4,
sum: Int = 50005000
```