Objectives

- (Practice) To gain experience using list comprehensions, zip (and unzip), and the fold and scan higher order functions on lists
- (Theory) To understand list comprehensions
- (Theory) To understand the fold and scan functions

Resources

You should refer to the following resources accessible via Blackboard:

- Topic 7 Lecture videos 7A – 7D
- **Topic 7 folder** (zipped) – includes the notes and slides for this topic
- External website for Scala doc and other **Learning Resources** (see the folder on Blackboard).

Introduction

The exercises in this topic cover three main topics:

1. List comprehensions (or for comprehensions as they are called in Scala). These are reminiscent of set comprehensions from set theory and allow the construction of lists using a generator, a filter, and a pattern. You should see how list comprehensions are alternative ways of writing what can otherwise be expressed by combining higher order functions. List comprehensions provide an alternative way of writing your code which, in some circumstances, can make it more readable.

2. Zipping lists. As the name implies, zipping two lists is the process of traversing two lists simultaneously and pairing the respective elements. If you think about how a zip works on a jacket, for example, then the "teeth" of the zip represent the two lists (one either side) and the action of zipping up the jacket pairs up the teeth. It is useful to be able to apply a function to the pairs so generated and this approach will be seen. Finally, if it is possible to zip two lists then it should be possible to unzip a list (of pairs) and recover the original lists. There are issues to consider here about the relative lengths of the two lists – what happens if one list is longer than the other? The zipAll method provides a means of padding the shorter list with default values.

3. Folding lists. There are many operations on lists that appear to be different from each other on first inspection, but are actually variations on the same pattern of computation. It is this pattern that is represented by the foldLeft and foldRight higher order functions. In fact, these are so general that many other list methods can be expressed using these patterns. You will also look at the scanLeft and scanRight functions which keep all the partial computations and return them in a list.

Activities

3.1 Watch the videos

Watch the videos 7A, 7B, 7C, and 7D. The accompanying slides can be found in the topic-7-folder. These videos give you an introduction to using list comprehensions; combining lists with zip; and reducing lists using fold and scan.

3.2 Install and run the ListDemo programs

Open your Scala IDE and within the **src** folder move to the **demo.list** package

Copy the Scala files **ListDemo3.scala** and **ListDemo4.scala** into the **list** package. The exercises are embedded within these programs. Follow the instructions in the comments.