Advances in Programming Languages

APL5: Coursework Assignment

Ian Stark and David Aspinall

School of Informatics
The University of Edinburgh

Monday 26 January 2009
Semester 2 Week 3

Outline

Outline

Aims of exercises

The aims of the homework exercises set in lectures include:

- To review lecture material
- To give some context for understanding the lectures
- To provide other sources and views on the lecture topic
- To help with learning by exploring the subject

Crucially, these effects arise from doing the exercises. They are not assessed, nor would this necessarily be useful: they are intended to be self-validating (i.e. you can tell when you have succeeded).

Although your coursework reports will be assessed, most of this also applies there: the purpose is for you to find out and learn new things.

Outline

Information flow in Jif

The *Jif* compiler extends the Java language with annotations for static analysis of security properties relating to the flow of information.

These annotations describe restrictions on how information is to be used: which *principals* control which information, and what they trust other principals to do with it. This gives increased assurance that trusted and untrusted information is used only according to explicit security policies.

Deputy

Memory annotations in Deputy

The *Deputy* project at Berkeley is developing a C compiler that can prevent a number of common programming errors.

In particular, Deputy provides type annotations with which programmers can describe the intended behaviour of pointers. The compiler will then apply suitable static and run-time checks to make sure these intentions are satisfied.

Haskell STM library

Software Transactional Memory in Haskell

The *STM* library for the Glasgow Haskell Compiler (GHC) provides high-level language support for coordinating concurrent computation, where multiple threads act simultaneously on shared datastructures.

Remarkably, STM does this without using locks. Instead, it uses efficient and optimistic *software transactions*, giving freedom from deadlock and promoting non-interfering concurrency. These transactions are modular and composable: small transactions can be glued together to make larger ones. Moreover, implementing this within the Haskell type system gives static guarantees that transactions are used correctly.

Hadoop

Petascale computing with Hadoop

The software platform *Hadoop* implements *MapReduce*, a programming model for massively distributed computation introduced by Google.

MapReduce splits input into pieces, applies a function in parallel to each piece, then merges the results together. Using a distributed file system, the framework can compute across thousands of machines, using redundancy and recovery mechanisms. MapReduce operations are programmed using Java and the Hadoop libraries, but the more abstract language *Pig Latin* can be used to compiled to Hadoop instead.

Multiple inheritance

Multiple inheritance in Scala with traits and mixins

The Scala language provides *traits* and *mixins* as modularisation constructs.

Mixin composition solves the infamous *multiple inheritance* ambiguity problem: does a class A that inherits from B and from C implement A.m as B.m or C.m? Java forbids multiple inheritance but provides interfaces. However, interfaces cannot contain implementations, leading to code duplication. Scala's trait and mixin constructs remedy this.

Outline

Dates and submission

Week 3 Monday 26 January: Topic announcement

Week 5 Monday 9th February: Assignment review lecture

Week 5 Friday 13 February: Intermediate report

Week 8 Friday 6 March: Final report

Each report should be submitted electronically as a PDF document. The recommended method for creating these is pdflatex with the article document class.

In addition, OpenOffice is freely available for Windows and Linux, installed on Informatics machines, and can write PDF. Mac OS X natively creates PDF. Microsoft provide PDF output as a plugin for Word 2007.

Dates and submission

Intermediate report

This document should contain:

- Your student number;
- The topic you have chosen;
- Three suitable references, which you have read; and
- A screenshot by you of the selected system in action.

One reference must be to a published paper; the other two may be too, but could also be white papers, web tutorials, manuals, or similar. In all cases provide enough information for someone else to obtain the document.

To create the screenshot, you will need to have your chosen system downloaded, installed, and running on a suitable machine.

Final report

This will be marked by the lecturers, and contributes 20% towards your grade for this course.

Assignment review

Week 5 Monday 9th February: Assignment review

A lecture to review the assignment topics.

We will discuss some of the topics in more detail. It is intended as a checkpoint, to see that you have made progress and been able to try out some programming for your chosen topic by this time.

It will also be an opportunity to discuss the final report content and look at some exemplars.

Suggested outline

Heading	Title, date
Abstract	This report describes ...
Introduction	Content summary, overview of report structure
Context	The problem domain
⟨Main topic⟩	What it is, how it works, advantages and limitations
Example	Annotated code, explanation, screenshot
	Salt: the example must in some way concern cooking or catering (e.g., recipe databases, canteen inventory, . . .)
Resources	For each notable resources used (article, tutorial, manual), give a summary in your own words of what it contains
Related work	Other approaches to the problem
Conclusion	What ⟨topic⟩ does, good and bad points
Bibliography	Full references for all resources used

Total 8–10 A4 pages. See the course web pages for further details.

Outline

Plagiarism

University of Edinburgh
Undergraduate Assessment Regulations 2008/09

Regulation 14

14.1 Plagiarism is the act of copying or including in one's own work, without adequate acknowledgement, intentionally or unintentionally, the work of another.

http://www.acaffairs.ed.ac.uk/Regulations/Assessment/08-09/UG.htm#Reg14

See also:

- University guidance
  http://www.acaffairs.ed.ac.uk/Administration/GuidanceInformation/AcademicBestPractice/Plagiarism/Index.htm
- Informatics policy
  http://www.inf.ed.ac.uk/teaching/plagiarism.html

Suitable working practices

Working practices

- Start with a blank document; all the words must be yours.
- Do not cut and paste from other documents.
  - Except for direct quotations, which must have source declared.
- Do not let others read your text; nor read theirs.

Aims of this assignment

- To learn about the chosen topic
- To improve researching and learning skills
- To demonstrate said knowledge and skills

The tangible outcome is a document, composed and written by you, demonstrating what you have learnt.

Outline

Summary

Topic choices

- Information flow in Jif
- Memory annotations in Deputy
- Software Transactional Memory in Haskell
- Petascale computing with Hadoop
- Multiple inheritance in Scala with traits and mixins

Coursework and learning

- Lecture exercises are there to be done
- Note the essay plan
- All your own work
- The aims of the coursework are to support learning