

G54FOP/FPP 2012/13

Mathematical Foundations of Programming & Mini-Project

Lecture 1-A

Administrative Details and Introduction

Henrik Nilsson

University of Nottingham, UK

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.1/13

Aims of G54FOP

- To provide a sound basis in a range of topics in the foundations of programming languages, including aspects of recent and current research. Specifically:
 - Basic lambda calculus
 - Operational semantics
 - Denotational semantics
 - Types and type systems
 - Domain theory

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.4/13

Organization (2)

- **Assessment:**
 - **G54FOP:**
 - Unseen 2-hour written examination: 100 %
 - **G54FPP:**
 - Written 10-page (3000–4000 words) report: 60 %
 - Presentation: 30 %
 - Participation in class discussions during presentations: 10 %

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.7/13

Finding People and Information

- Henrik Nilsson
Room A08, Computer Science Building
e-mail: nhn@cs.nott.ac.uk
tel: 0115 846 6506
- Main module web page (both G54FOP & G54FPP):
www.cs.nott.ac.uk/~nhn/G54FOP

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.2/13

G54FPP: Optional Mini-Project

- Can only be taken together with G54FOP
- Aim is to provide G54FOP students with the opportunity to deepen their understanding by an in-depth study of a specific topic related to G54FOP.
- Non-exclusive list of suggested topics available via the module web page; additional topics or amended versions can be discussed.

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.6/13

Organization (3)

- **Resit assessment:**
 - **G54FOP:** Unseen 2-hour written examination (like first sit)
 - **G54FPP:** Extended, 20-page (7000–8000 words) report.

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.8/13

Contacting Me

- I will be available immediately after each lecture for course-related matters.
- E-mail.
- Make an appointment if necessary.

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.3/13

Organization (1)

- **G54FOP:** Two lectures per week:
 - Thursdays, 11–12, B12, AMEN
 - Fridays, 11–12, B12, AMEN
- **G54FPP:**
 - Written report on a research article/topic.
 - Report deadline: Friday 3 May, 17:00 (CW submission)
 - Presentation (20 + 5 minutes).
 - Presentations towards end of spring semester (2 per lecture); spare G54FOP slots or specially scheduled.

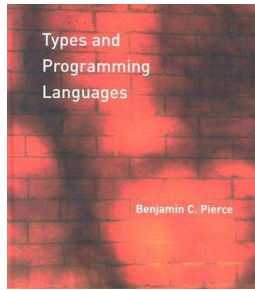
G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.9/13

Literature

- Your own notes from the lectures!
- The lectures will draw from:
Benjamin C. Pierce. *Types and Programming Languages*, MIT Press, 2002.
A good reference for large parts of the module.
- Other texts on lambda calculus, semantics, etc. I'll make some suggestions as we go along.

G54FOP/FPP 2012/13 Mathematical Foundations of Programming & Mini-Project Lecture 1-A – p.9/13

Literature (2)



G54FOP/FPP 2012/13Mathematical Foundations of Programming & Mini-ProjectLecture 1-A - p.10/13

Content (1)

- Mathematical preliminaries: formal languages, grammars, induction, inference rules, ...
- Semantics:
 - Operational
 - Denotational
- Lambda calculus
- Type theory
 - Simply-typed lambda calculus
 - Recursive types?
 - Polymorphism (System F)

G54FOP/FPP 2012/13Mathematical Foundations of Programming & Mini-ProjectLecture 1-A - p.11/13

Content (2)

With G54FOP as a basis, G54FPP allows you to for example study topics like the following:

- Program analysis
 - Type reconstruction
 - Type and effects systems
 - Abstract interpretation
 - Data-flow analysis
- Program logics and correctness
 - Hoare logic
 - Calculation of programs from specifications

G54FOP/FPP 2012/13Mathematical Foundations of Programming & Mini-ProjectLecture 1-A - p.12/13

Your background?

- Discrete mathematics, sets, logic?
- Formal languages, grammars, abstract syntax?
- Mathematical induction?
- Structural induction?
- Functional programming? Haskell?
- Semantics?
- Type theory?

G54FOP/FPP 2012/13Mathematical Foundations of Programming & Mini-ProjectLecture 1-A - p.13/13