Category Theory and Coalgebra: Lecture 1

Jurriaan Rot

Radboud University, 2020

iCIS | Software Science
Radboud University

Introducing the lecturer(s)

- Jurriaan Rot — jrot@cs.ru.nl

  http://jurriaan.me
- Bart Jacobs — bart@cs.ru.nl

https://www.cs.ru.nl/B.Jacobs/

Website

http://www.cs.ru.nl/~jrot/CTC20/

Course information (lecture dates, exercises, literature, announcements, ...) on the **website**

Please also register for the course (email announcements and grades via Brightspace)

Time and date

Lectures

- **Mondays** 13:30 - 15:15, HG00.086
- Weeks 6-12 and 16-22 (some cancelled; 12 lectures total)
- Schedule announced on the website

Exercise class

- **Wednesdays** 10:30 - 12:15, MERC I 00.28
- **No class this Wednesday (Feb 5)**
- Meant to **practice** and **ask questions**: no new material

Literature

We'll use several sources, in particular research papers; announced on the website.

Easy introduction to coalgebra:

Jan Rutten. The method of coalgebra: exercises in coinduction.
http://www.cs.ru.nl/~jrot/coalg18/course.pdf

We will also sometimes use:

Bart Jacobs. Introduction to coalgebra, version 2.0, 2012.
http://www.cs.ru.nl/B.Jacobs/CLG/JacobsCoalgebraIntro.pdf
Official version published by Cambridge University Press, 2016.

Second half (mostly): lecture notes

Assessment

- Two graded assignments: first in March/April, second in May/June (separate from the homework in the exercise classes, which is not graded)
- A final exam (June 22nd, 2020), resit in July
- Final grade: (H+E)/2, where H is the grade given for the homework assignments and E is the grade given for the final exam

Category theory

- Formalism/language/toolkit offering a really general and unifying perspective on common phenomona
- Used in various areas of mathematics and theoretical computer science
- Examples in computer science:
  - functional programming / computation
  - coalgebra, of course!
  - compositionality (see "applied category theory" hype)
  - ...
- Fundamental: offers a new way of thinking!

Coalgebra

- Mathematical theory of **state-based systems**
- Uniform study of many types of systems and models; various kinds of automata, (infinite) data structures such as sequences and trees, probabilistic systems, ...
- Parametric in the type of system, which gives, for each type, a notion of **behavioural equivalence** and **semantics**, and associated **proof methods**
- Applications, fundamental insights, new perspectives

iCIS | Software Science
Radboud University

Colgebra: small bit of history

- Machines and data structures in category theory (70's, 80's)
- Behavioural equivalence of processes (Milner & Park, 80's)
- Processes and non-wellfounded set theory (Peter Aczel (late 80's))
- Universal coalgebra (Jan Rutten, Bart Jacobs, . . . – from 90's on)
- By the end of the 90's: active and diverse area of research, conference & workshop of it's own

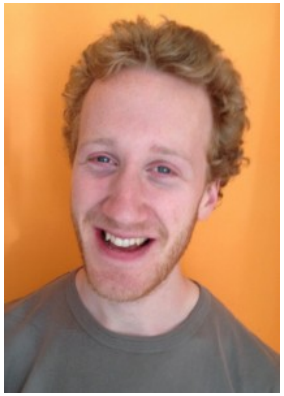
Coalgebra research today

Active community studying the fundamental theory of coalgebras, as well as applications in/connections to:

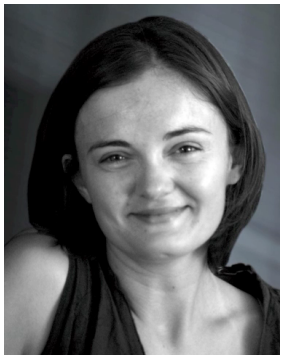
- automata theory
- programming language semantics
- concurrency theory
- probabilistic systems
- modal logic
- theorem proving
- type theory
- ...

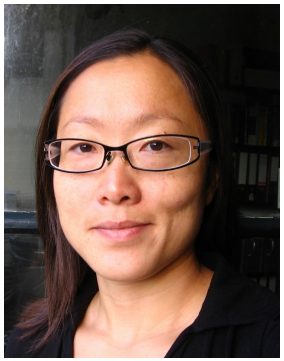
Coalgebra in Nijmegen





and people who now work elsewhere:





iCIS | Software Science
Radboud University

What will you learn

In this course:

- basic category theory
- theory of coalgebras — modelling state-based systems
- applying the coalgebraic notions of behavioural equivalence and coinduction
- the interplay between algebra and coalgebra, induction and coinduction
- using coalgebraic/coinductive proof techniques;
  - *highlight: efficient algorithm for equivalence of non-deterministic automata*
- probabilistic computation and monads

Overview

Basic theory of coalgebras and category theory (5 lectures)

- streams and coinduction
- categories, functors, natural transformations
- coalgebras

Coinduction: applications and advanced topics (4 lectures)

- relation lifting and bisimulation
- coinduction
- algorithms for equivalence and minimisation of automata

Monads and probability (3 lectures)

- probabilistic computation
- monads