

Algebraic and Coalgebraic Models in Software Development

Manuel A. Martins, UA

Dirk Hoffman, UA

Luis S. Barbosa, UM

MAP-i

Universidade de Aveiro

28 September, 2009

Maths in Computer Science?

Excerpt from the CAMBRIDGE EVENING NEWS (April 2002):

Computer Scientist Gets to the "Bottom" of Financial Scandal

A Cambridge computer professor, Simon Peyton Jones, has made an interesting discover regarding the Enron collapse. (...) Enron's collapse was due to a nearly impenetrable web of financial contracts that disguised the true financial state of the company (...)

(...) Accountants find that even when they are scrupulously honest about the valuation of such contracts there can still be sharp disagreements in regard to the worth of trading reserves, debts, and other components.

Maths in Computer Science?

Enter Professor Peyton Jones. As part of his research at Microsoft in Cambridge, he developed a computer language for describing and valuing financial contracts (...)

With colleagues Jean-Marc Eber and Julian Seward, they developed a language capable of accurately describing and valuing even the most complex financial instruments.

"While accountants find financial derivatives to be mysterious and difficult, for us they are just ordinary recursive equations", says Jones.

Maths in Computer Science?

Notice the "magic words"

- ... *a language capable of accurately describing and valuing ...*
- ... *just ordinary recursive equations ...*

In other words:

- a mathematical description language for **modelling** ...
- ... and a calculus for **reasoning about models**

i.e.,

... the two ingredients of what is called **Formal Methods** in CS

actually,

the hallmark of any **true Engineering discipline**

Maths in Computer Science?

Notice the "magic words"

- ... *a language capable of accurately describing and valuing* ...
- ... *just ordinary recursive equations* ...

In other words:

- a mathematical description language for **modelling** ...
- ... and a calculus for **reasoning about models**

i.e.,

... the two ingredients of what is called **Formal Methods** in CS

actually,

the hallmark of any **true Engineering discipline**

Maths in Computer Science?

Notice the "magic words"

- ... *a language capable of accurately describing and valuing* ...
- ... *just ordinary recursive equations* ...

In other words:

- a mathematical description language for **modelling** ...
- ... and a calculus for **reasoning about models**

i.e.,

... the two ingredients of what is called **Formal Methods** in CS

actually,

the hallmark of any **true Engineering discipline**

Maths in Computer Science?

IT became ubiquitous to modern life long before a **solid scientific methodology**, let alone formal foundations, has been put forward

To be contrasted with

school physics' **problem-solving strategy**:

- understand the problem
- build a mathematical **model** of it
- animate and **reason** in such a model
- upgrade the model whenever necessary
- calculate a **solution** and implement it

Maths in Computer Science?

IT became ubiquitous to modern life long before a **solid scientific methodology**, let alone formal foundations, has been put forward

To be contrasted with

school physics' **problem-solving strategy**:

- **understand** the problem
- **build** a mathematical **model** of it
- **animate** and **reason** in such a model
- **upgrade** the model whenever necessary
- **calculate** a **solution** and **implement** it

Information & Behaviour

Computer Science deals with both

information and behavioural structures

This course discusses semantics of computation

- resorting to the algebra vs coalgebra duality as a mathematical 'explanation' of the intuitive symmetry between data and behavioural structures;
- at both levels: foundations and development methods

Information & Behaviour

Computer Science deals with both

information and behavioural structures

This course discusses semantics of computation

- resorting to the algebra vs coalgebra duality as a mathematical 'explanation' of the intuitive symmetry between data and behavioural structures;
- at both levels: foundations and development methods

Information & Behaviour

(Initial) **algebras** are abstract description of **data structures**

$$[\text{nil}, \text{cons}] : \mathbf{1} + O \times L \longrightarrow L$$

In general:

a **tool box**:



an **assembly process**:



artifact \xrightarrow{a} artifact

The emphasis is on **construction**

Information & Behaviour

(Final) **coalgebras** are abstract descriptions of **systems' behaviours**

$$\langle \text{at}, \text{m} \rangle : U \rightarrow O \times U$$

In general:

a **lens**:

$$O \sim O$$

an **observation structure**: universe \xrightarrow{c} $O \sim O$ universe

The emphasis is on **observation**

Course Plan

1. **Introduction to category theory for computer science**
2. **Algebras and algebraic specification**
 - 2.1 Signatures, models, morphisms
 - 2.2 Equational logic
 - 2.3 Refinements
 - 2.4 Introduction to the theory of institutions
 - 2.5 Behavioural specifications
 - 2.6 Tools: [OBJ](#), [CASL](#)
 - 2.7 Applications: [fast-evolving requirements specification](#)
3. **Coalgebras and coalgebraic specification**
 - 3.1 Coalgebras
 - 3.2 Bisimulation, coinduction and final coalgebras
 - 3.3 Logics for coalgebras
 - 3.4 Tools: [CIRC](#)
 - 3.5 Applications: [software components and services](#)

Lecturers

- Manuel António Martins, Maths, UAveiro
- Dirk Hoffman, Maths, UAveiro
- Luís Soares Barbosa, CS, UMinho

... and a number of [invited lecturers](#), among whom

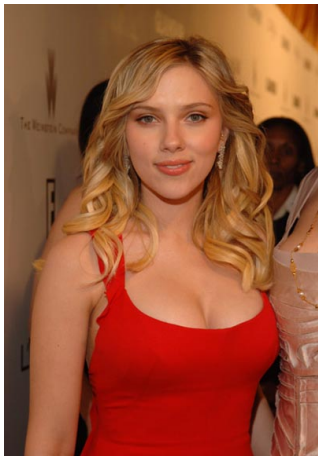
Lecturers

- Manuel António Martins, Maths, UAveiro
- Dirk Hoffman, Maths, UAveiro
- Luís Soares Barbosa, CS, UMinho

... and a number of [invited lecturers](#), among whom

Lecturers

- ...
- Scarlett Johansson (scientific approval pending)



So ...

Why haven't you registered YET?

Context

MONDRIAN

PTDC/EIA-CCO/108302/2008

- on foundations of service-oriented computing
- possible grants

Industrial links

(e.g., BDE with top software companies)

Context

MONDRIAN

PTDC/EIA-CCO/108302/2008

- on foundations of service-oriented computing
- possible grants

Industrial links

(e.g., BDE with top software companies)

Why haven't you registered YET?

We seek **rigour** and **simplicity** in software design

There are two ways of constructing a software design: one way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies.

C.A.R. Hoare

... following a **sound** path

Nothing is more practical than a good theory

V.I. Lenine

Our solutions should be as simple as possible, but no simpler

Albert Einstein

Why haven't you registered YET?

We seek **rigour** and **simplicity** in software design

There are two ways of constructing a software design: one way is to make it so simple that there are obviously no deficiencies, and the other way is to make it so complicated that there are no obvious deficiencies.

C.A.R. Hoare

... following a **sound** path

Nothing is more practical than a good theory

V.I. Lenine

Our solutions should be as simple as possible, but no simpler

Albert Einstein