Mathematical Semantics of Computer Systems, *MSCS* (4810-1168)
Handout for Lecture 1 (2016/9/26)

Ichiro Hasuo, Dept. Computer Science, Univ. Tokyo
`http://www-mmm.is.s.u-tokyo.ac.jp/∼ichiro`

1 Today's Lecture

Introduction. Overview of the topics covered. "Background check." Some first notions in category theory.

2 On This Course

Below is an excerpt from the course's website

> `http://www-mmm.is.s.u-tokyo.ac.jp/~ichiro/`,
> → "TEACHING" → "Mathematical Semantics of Computer Systems"

Beware: **what is on the website is up-to-date and correct.**

2.1 General

- Monday 10.25h-12.10h
- Room 102, School of Science Bldg. No. 7
- For Graduate Students. Code: 4810-1168
- Lectures are in English

2.2 Course Objective

Correctness of computer systems and programs is a pressing issue in the modern world. Their verification—providing mathematical proofs for their correctness—calls for their mathematical modeling; this is what the field called semantics is all about. The course introduces some basic techniques in the field, most notably those which employ category theory.

Keywords: category theory, lambda calculus, type theory, logic, categorical semantics, algebraic semantics, denotational semantics, algebra, coalgebra, automaton, fibration, categorical logic

2.3 References

- [Lambek & Scott 1986]

  J. Lambek, P. J. Scott, Introduction to Higher-Order Categorical Logic. Cambridge University Press, 1988
- [Awodey 2010]

  S. Awodey, Category Theory. Oxford University Press, 2010.
- [Leinster 2014]

  T. Leinster, Basic Category Theory. Cambridge University Press, 2014.

- [Jacobs 2012]

  Bart Jacobs, Introduction to Coalgebra: Towards Mathematics of States and Observations. Draft of a book, available online. 2012

- [Jacobs, 1999]

  Bart Jacobs, Categorical Logic and Type Theory. Elsevier Science, 1999

2.4 Course Structure

Part I: Basic Category Theory

We exhibit basic constructs in category theory used in the subsequent parts.

- categories, functors, natural transformations
- monoids and preorders as degenerate cases
- epis and monos
- (co)products, (co)equalizers, (co)limits
- adjoint functors, equivalences of categories
- exponents, Cartesian closed categories

We will roughly follow [Lambek & Scott 1986, Part 0] which is however way too fast for beginners; [Awodey 2010, Leinster 2014] will be a nice supplementary material.

Goal of Part I Familiarize yourself with the Yoneda lemma, a celebrated result in category theory by Prof. Nobuo Yoneda (Dept. IS, U. Tokyo), desirably in its end/coend form.

The Curry-Howard-Lambek Correspondence

As a first example of categorical semantics, a correspondence between typed lambda-calculi and Cartesian closed categories is introduced. An emphasis is on the view that categorical semantics is an extension of algebraic semantics (such as Boolean algebras for classical logic and Heyting algebras for intuitionistic logic).

- typed lambda-calculi
- natural deduction for intuitionistic logic
- syntactic presentation of Cartesian closed categories
- free Cartesian closed categories and completeness

We will follow [Lambek & Scott 1986, Part 1] omitting some topics.

Goal of Part II The completeness result for Moggi's computational lambda-calculus as an extension of the Curry-Howard-Lambek correspondence, following

> Computational lambda-calculus and monads. Tech. Report ECS-LFCS-88-66, Edinburgh Univ., 1988.

Part III: Coalgebras and State-Based Transition Systems

Our second usage of category theory has quite different a flavor: it models various state-based transition systems as coalgebras, a categorical dual of algebras. Here the view of categories as ”many-sorted algebras (or monoids)” is not so important as in the previous part—in fact we will almost exclusively work in the category of sets and functions. An alternative categorical genericity from which we benefit is a functor as a parameter: it allows to develop a uniform theory of state-based systems like automata of various kinds (deterministic, nondeterministic, probabilistic, or even quantum).

- coalgebraic modeling
- bisimulation
- final coalgebras
- (process algebra and bialgebra)

We will follow [Jacobs 2012, Chapters 1-3].

Goal of Part III Familiarize yourself with *coinduction as a proof principle.* Prove some stream equalities with the principle.

Part IV: Fibrations and Categorical Logic

The last part introduces a more advanced bit of category theory, namely the theory of fibrations that are a categorical way of “organizing indexed entities.” It will be applied to modeling predicate logic, extending Part 2. If time allows another application, namely coinductive predicates in state-based transition systems, will be introduced. This will unify the views in Parts 2–3 and illustrate the versatility of the mathematical language of category theory.

- fibrations
- indexed categories, the Grothendieck construction
- fiberwise structures
- (co)products between fibers
- equational logics as fibrations
- first-order predicate logics as fibrations
- (inductive and coinductive predicates)

The topics are picked from [Jacobs, 1999]

Goal of Part IV Understand coinduction up-to formulated in a fibration, as in

> Filippo Bonchi, Daniela Petrisan, Damien Pous, Jurriaan Rot: Coinduction up-to in a fibrational setting. CSL-LICS 2014: 20

3 Today's Agenda

3.1 Categories, Functors

Definition. Category, object, arrow (= morphism).

Example. **Sets**; a preorder as a category; a monoid as a category; the category **Mon** of monoids and monoid homomorphisms.

Remark 1. To familiarize yourself with the following abstract categorical notions, always try to think of their instances when the category in question is one of the previous examples.

Remark 2. An "abstract" definition of category by a deductive system, in [Lambek & Scott]

Definition. Proper class, small set, small category, locally small category

Definition. Functor

Example. Monotone functions as functors (for preorders considered as categories); monoid homomorphisms as functors (for monoids).

Example. Graphs as functors. Monoid/group actions as functors.

3.2 Reasoning with Arrows

Definition. Epi, mono. Split epi, split mono.

Definition. Product, coproduct.