CMPT 863 Programming Language Semantics
2013-2014 Term 2

Practical Foundations of (Constructive) Mathematics and Category Theory

Christopher Dutchyn

December 29, 2013

Computing Science is mathematical science. Fundamental to all problem-solving is the ability to construct solutions, and verify their correctness. Computer science uses formal systems to formulate problem solutions (aka programs) and to analyse and verify them. This course focuses on the mathematical tools, primarily constructive logic and category theory, used to describe the semantics of programming languages. This firm foundation for program construction and verification supports our most important class of programs: programming language implementations. Indeed, the current trend in programming languages research is toward increasingly rigorous mechanized metatheory: "*Soon, papers will be rejected from POPL because they didn't include mechanically-checked proofs of the properties claimed.*"

This course introduces the students to

- *proof theory*: proofs as formal objects of study, including
  - natural deduction and sequent styles
  - logical connectives as proof combinators
  - rules of reasoning as proof transformations
  - automated and semi-automated proof construction
- *type theory*: types as specifications of programs, including
  - mapping inductive datatypes into symbolic logic formulae
  - mapping program statements into proof term transformations,

  leading to the *Curry-Howard* correspondence: programs as proofs of well-formed propositions via type analysis

- *partially-ordered sets and lattices*:
  - for characterizing recursion and fixed points (and hence iterative and non-terminating programs) using *Scott-Ershov* domains,
  - leading to static analysis and approximation, especially in the formulation of projection/concretization pairs as *Galois* connections
- and ending with the underlying *category theory*, yielding an insight into
  - monoids as programming constructs,
  - closed cartesian categories as models of lambda-calculus and logic
  - *Floyd-Hoare* type theory for showing formal correctness.

1 Textbooks

Most of the material comes from the first four chapters of the textbook, available online. The textbook will also be on reserve in the library, to access the missing figures.

1.1 Required

- Paul Taylor, *Practical Foundations of Mathematics*, Cambridge Studies in Advanced Mathematics, Cambridge University Press, 1999; text available online at `http://www.paultaylor.eu/~pt/prafm/`.

1.2 Supplemental

- Joachim Lambek and Phillip J. Scott, *Introduction to Higher-Order Categorical Logic*, Springer, 1994.
- Benjamin Pierce, *Basic Category Theory for Computer Scientists*, MIT Press, 1991.
- Robert F. C. Walters, *Categories and Computer Science*, Cambridge University Press, 1991.

2 Staff

Instructor: Chris Dutchyn

email `mailto:dutchyn@cs.usask.ca`

web `http://www.cs.usask.ca/faculty/cjd032`

office Thorvaldson 281.10

3 Meetings

Discussion/lecture on Monday, Wednesday, Friday from 1:30 to 2:20 in Spinks 386.

4 Grading

Regular attendance and productive classroom participation is mandatory for this reading-group-like course. There will be four assignments, drawn primarily from the textbook. In addition, the students will be asked to present assignment solutions, to encourage classrom participation and develop skill in communicating deep technical material to a well-prepared audience. The final exam will be a two-hour oral exam, scheduled at the mutual convenience of the student and instructor.

Table 1: CMPT 863 Marking

Item	Description	Weighting
Assignments	(four)	60%
Attendance		10%
Final	(oral)	30%
Total		100%

4.1 Sitting In

Students cannot unofficially audit (sit-in on) the meetings. Because much of the course entails in-class discussion, anyone sitting-in would complete the substantive work of the course, and merits credit for that effort.

- *Dec 29, 2013: initial release as CMPT 863*