## Constructive type theory

This course of three lectures will offer an introduction to Martin-Löf's constructive type theory, with an emphasis on basic ideas and concepts rather than technicalities. Although some use of mathematical symbolism and technical terminology is inevitable when dealing with type theory, I will seek always to motivate the various parts of the theory dealt with. The course should be accessible to anyone who knows the syntax and intended interpretation of predicate logic; knowledge of Gentzen-style natural deduction will be useful, but is not necessary. Topics to be covered include basic concepts such as judgement (both categorical and hypothetical), proposition, set, and type; the definition of the primitive constants of the language by means of introduction- and elimination-rules; and the hierarchy of higher types.

Constructive type theory involves a rather different way of thinking about logic from what most logicians and students of logic will be used to. For instance, in contrast to the languages of standard predicate logic, constructive type theory is an interpreted formalism. It is a language designed in order to reason with and not as a language to reason about. There is therefore no clean distinction between syntax and semantics in the sense of metamathmatics, since what is the syntax of a symbol depends on its semantics, and vice versa. Moreover, constructive type theory rests on a distinction between proposition and judgement that may seem mysterious the first time one sees it. I will do my best to point out and explain these various discrepancies with standard thinking about logic.

**Tentative outline.**

8 Oct General introduction; the notions of judgement and proposition; the notions of type and set; the hierarchy of higher types.

15 Oct The Curry-Howard isomorphism; formation-, introduction-, elimination-, and identity-rules; the logical constants.

22 Oct More constants: the natural numbers, finite sets, propositional identity.

**References.**

Most of what I will discuss is treated in

- Martin-Löf, P.: *Intuitionistic Type Theory* (Naples 1984: Biblipolis).

The book can be downloaded from `https://github.com/michaelt/martin-lof`. The hierarchy of higher types was being developed just around the time this book was published, hence it is not presented there. A brief presentation can be found in chapter 8 of

- Ranta, A: *Type-Theoretical Grammar* (Oxford 1994: Oxford University Press).

A fuller, though more technical, presentation can be found in

- Nordström, B., Petersson, K., Smith, J.M.: Martin-Löf's Type Theory. In *Handbook of Logic in Computer Science* (Oxford 2000: Oxford University Press).

Another useful textbook is

- Nordström, B., Petersson, K., Smith, J.M.: *Programming in Martin-Löf's Type Theory* (Oxford 1990: Oxford University Press).

The two last items can be downloaded from `http://www.cse.chalmers.se/~bengt/`. In motivating the need for a distinction such as that between judgements and propositions I will make use of considerations found in

- Sundholm, B.G.: Semantic values for natural deduction derivations. *Synthese* 148:623–638.

The type-theoretical way of thinking is nicely portrayed on the first pages of

- de Bruijn, N.G.: On the role of types in mathematics. *Cahiers du Centre de Logique* 8:27–54,

available at `http://alexandria.tue.nl/repository/freearticles/597627.pdf`.