On the PhD thesis "Reference and Computation in Intuitionistic Type Theory" by Johan Granström

Peter Dybjer

Chalmers University of Technology

Uppsala, 16 January, 2009

What is Intuitionistic Type Theory?

A (series of) formal logical theory(ies) introduced by Per Martin-Löf:

1972 (1997) *An intuitionistic theory of types*

1973 (1975) *An intuitionistic theory of types: predicative part*

1979 (1982) *Constructive mathematics and computer programming*

1980 (1984) *Intuitionistic Type Theory*, book published by Bibliopolis

1987 (?) *Philosophical implications of type theory*

1990 *Substitution calculus*

- A full-scale framework for constructive mathematics in the style of Bishop ("like ZF for intuitionism")
- A functional programming language with dependent types where all programs terminate

Origins of Intuitionistic Type Theory

- Intuitionism: Brouwer (1908), Heyting, Kolmogorov
- Type Theory: Russell (1908), Church (1940)
- Formulas as Types: Curry, Howard (1968)

Importance of Intuitionistic Type Theory: Constructive Mathematics = Computer Programming

- One of several foundational system for *predicative* constructive mathematics (others are constructive set theory, explicit mathematics, predicative topos)
- Formalization of (constructive) mathematics. Systems such as NuPRL, Coq (impredicative type theory), ALF, Lego (impredicative), Agda, Epigram, Matita, ...
  Look at `http://www.cs.ru.nl/~freek/100/index.html` (the proofs of 81 of 100 "top theorems" have been formalized)
- Functional programming languages with dependent types: "the search for the perfect programming language" (DML, Cayenne, Haskell with generalized algebraic data types, Agda, ...)

Meaning explanations: "direct", "intuitive" semantics as opposed to "metamathematical" semantics

The other foundational systems for *predicative* constructive mathematics (constructive set theory, explicit mathematics, predicative topos) can be interpreted in intuitionistic type theory (and vice versa), but type theory provides a meaning theory which extends and refines the Brouwer-Heyting-Kolmogorov interpretation of intuitionistic logic:

1979 (1982) *Constructive mathematics and computer programming*

1980 (1984) *Intuitionistic Type Theory*, book published by Bibliopolis

However, cf also

Siena 1983 *On the meaning of the logical constants and the justification of the logical laws*

for meaning explanations which are not based on type theory

Meaning according to "Constructive Mathematics and Computer Programming" (1979)

Technically, a variant of realizability interpretation (originally introduced by Kleene).

- Untyped terms; essentially lambda terms with constants, including terms denoting types.
- Computation relation $\Rightarrow$ between *closed* untyped terms; relates a term to its lazy value (canonical form = weak head normal form) where only outermost form is constructor.

Meaning explanations: pre-mathematical or meta-mathematical?

Martin-Löf: *Intuitionistic Type Theory*, Bibliopolis, 1984, p 1, par 1.

"Mathematical logic and the relation between logic and mathematics have been interpreted in at least three different ways:

1. *mathematical logic as symbolic logic, or logic using mathematical symbolism;*
2. *mathematical logic as foundations (or philosophy) of mathematics;*
3. *mathematical logic as logic studied by mathematical methods, as a branch of mathematics.*

We shall here mainly be interested in mathematical logic in the second sense. What we shall do is also mathematical logic in the first sense, but certainly not in the third."

Reference (wikipedia)

In semantics, *reference* is generally construed as the relationships between nouns or pronouns and objects that are named by them. Hence the word "John" refers to John. The word "it" refers to some previously specified object. The object referred to is called the referent of the word.

Gottlob Frege argued that reference cannot be treated as identical with meaning: "Hesperus" (an ancient Greek name for the evening star) and "Phosphorus" (an ancient Greek name for the morning star) both refer to Venus, but the astronomical fact that '"Hesperus" is "Phosphorus"' can still be informative, even if the 'meanings' of both "Hesperus" and "Phosphorus" are already known. This problem led Frege to distinguish between the *sense* and *reference* of a word.

The thesis of Johan Granström: summary

III-V A new version of Intuitionistic Type Theory. Inference rules and Meaning Explanations. Philosophical and historical remarks. Cf "Intuitionistic Type Theory" (Bibliopolis, 1984). Both formal theory and meaning explanations are modified (modernized, generalized)

I-II Background. Discussion of basic logical concepts from a philosophical and historical perspective.

VI Indirect proof, law of excluded middle. Certainty vs truth.

Syntax and inference rules compared with Bibliopolis theory

Judgement forms

- Judgement forms for canonical forms $c \, \varepsilon \, \mathrm{el}(C)$ in addition to the general form $a : \mathrm{el}(A)$ which allows non-canonical forms, Also $F \, \varepsilon \, \mathrm{fam}(C)$.
- Computation assertion (judgement) $\mathrm{el}(A) : a \Rightarrow c \, \varepsilon \, \mathrm{el}(C)$
- Substitution calculus incorporating ideas from categorical logic (categories with families)

Sets

- Inductive definitions of families of sets (vectors)
- "Eager" as well as "lazy" sets (data types)
- Quotient sets (e g type of arithmetic expressions under associativity)
- Coinductive definitions (streams)

Syntax and inference rules compared with Bibliopolis theory

No explicit discussion of

- generalized inductive definitions (wellorderings, etc)
- universes and inductive-recursive definitions
- identity set

Meaning explanations compared with Hannover

What's written between the lines? Is it a "clarification" or a "variation" of Hannover? Untyped computation system vs typed computation system.

- Untyped terms
- Computation relation between *closed* untyped terms

Rules of type theory are justified with respect to this: two levels.
Here terms and computation rules are part of the formal system

- Terms are introduced together with their types
- Typed computation judgement
- Canonical element judgement

Rules are divided into "meaning determining" (D), "justified" (J), and "recognized" (R).
Meaning theory is adapted to substitution calculus formulation and accounts for the new forms of sets discussed above. Eager vs lazy

Assessment

- A *monography* in the best sense! Well-written, highly readable! A comprehensive treatment of an important subject. Explains a version of Intuitionistic Type Theory "without interruption".
- Work in the junction of philosophy, mathematics, and computer science.
- Clarification or variation of meaning theory using "typed computation systems". Integrating modern developments (substitution calclus).
- Generalizations. Eager, coinductive, quotient.
- Interesting final chapter on classical logic, interpretations of excluded middle, etc.
- Extensive account of historical development of basic notions!