Logic Encodings in Dependent Type Theory

Florian Rabe

Part of the course on Computational Logic by Michael Kohlhase

Fall 2007, Jacobs University Bremen

26.11.2007

Meta-Languages

- Meta: after, beyond (Greek)
- Informally, M is the meta-language of the object language L if M is used to define (the syntax or semantics of) L or if M is used as the underlying framework for L.
- Examples:
  - philosophy – mathematics – physics – chemistry – biology
  - mathematics – assembler – bytecode – Java – a Java program
  - natural language – ZFC – FOL – rings
  - natural language – higher set theory – category theory – institutions – FOL – rings
  - natural language – type inference systems – LF – linear algebra
  - natural language – type inference systems – LF – FOL – rings

Foundations of Mathematics

- In mathematics, there is not a single uppermost meta-language.
- Rather, there are several competing ones, so-called foundations of mathematics.
- Mathematicians usually use set theory, but there are variants: e.g., ZF(C), von Neumann-Bernays-Gödel set theory, Tarski-Gröthendieck set theory; more expressive foundations are needed to talk about $\mathcal{C}\mathrm{at}$.
- (Mainly) computer scientists also use, e.g., simple and dependent type theory.
- These foundations are usually encodable in each other, e.g., STT – ZFC within the Isabelle system, DTT – DTT in LF, institutions – DTT.

Duplicate Role of DTT

- Thus DTT plays a duplicate role:
  - It is one of the formal systems we are interested in (just like FOL or STT), and in which we represent theories (such as rings or categories).
  - It is a framework in which we encode other formal systems.
- Principle of encoding a logic L in LF:
  - Give LF-signature Σ_L declaring all connectives, quantifiers, axioms, and proof rules of L.
  - A specific L-signature is an LF-signature of the form Σ_L, Σ.
  - Because of Curry-Howard, L-theories have the same form as L-signatures!

Examples for Logic Encodings in LF

Lecture, exercises

The Curry-Howard-correspondence for DTT

Principle:

- Connectives, quantifiers correspond to type constructors; formulas to types
- Proof rules correspond to term constructors; proofs to terms
- Extension to category theory via the category of proofs/contexts

Logic	Type Theory	Category Theory
truth	unit type	terminal object
falsity	void type (always emtpy)	initial object
conjunction	product types	product objects
disjunction	union types	coproduct objects
implication	function types	exponential objects

The Curry-Howard-correspondence for DTT

Principle:

- Connectives, quantifiers correspond to type constructors; formulas to types
- Proof rules correspond to term constructors; proofs to terms
- Extension to category theory via the category of proofs/contexts

Logic	Type Theory	Category Theory
truth	unit type	terminal object
falsity	void type (always emtpy)	initial object
conjunction	product types	product objects
disjunction	union types	coproduct objects
implication	function types	exponential objects
universal quantifier	dependent function type	

The Curry-Howard-correspondence for DTT

Principle:

- Connectives, quantifiers correspond to type constructors; formulas to types
- Proof rules correspond to term constructors; proofs to terms
- Extension to category theory via the category of proofs/contexts

Logic	Type Theory	Category Theory
truth	unit type	terminal object
falsity	void type (always emtpy)	initial object
conjunction	product types	product objects
disjunction	union types	coproduct objects
implication	function types	exponential objects
universal quantifier	dependent function type	
existential quantifier	dependent product type	

The Curry-Howard-correspondence for DTT

Principle:

- Connectives, quantifiers correspond to type constructors; formulas to types
- Proof rules correspond to term constructors; proofs to terms
- Extension to category theory via the category of proofs/contexts

Logic	Type Theory	Category Theory
truth	unit type	terminal object
falsity	void type (always emtpy)	initial object
conjunction	product types	product objects
disjunction	union types	coproduct objects
implication	function types	exponential objects
universal quantifier	dependent function type	right adjoint to substitution
existential quantifier	dependent product type	left adjoint to substitution

Curry-Howard-correspondence in LF

- Type A is inhabited iff there is a term of type A.
- Types corrrespond to judgments.
- Inhabitation of a type A corresponds to "A holds".

Curry-Howard-correspondence in LF

- Type A is inhabited iff there is a term of type A.
- Types corrrespond to judgments.
- Inhabitation of a type A corresponds to "A holds".
- $a : \Pi x_1 : A_1 . \ldots . \Pi x_n : A_n . type$ — a family of judgments
- base judgment: $a\ t_1\ \ldots\ t_n$ — "a of t_i (is inhabited)"

Curry-Howard-correspondence in LF

- Type A is inhabited iff there is a term of type A.
- Types corrrespond to judgments.
- Inhabitation of a type A corresponds to "A holds".
- $a : \Pi x_1 : A_1 . \dots . \Pi x_n : A_n . type$ — a family of judgments
- base judgment: $a\ t_1\ \dots\ t_n$ — "a of t_i (is inhabited)"
- hypothetical judgment: $A \rightarrow B$ — "if A (is inhabited), then B (is inhabited)"

Curry-Howard-correspondence in LF

- Type A is inhabited iff there is a term of type A.
- Types corrrespond to judgments.
- Inhabitation of a type A corresponds to "A holds".
- $a : \Pi x_1 : A_1 . \ldots . \Pi x_n : A_n . type$ — a family of judgments
- base judgment: $a\ t_1\ \ldots\ t_n$ — "a of t_i (is inhabited)"
- hypothetical judgment: $A \to B$ — "if A (is inhabited), then B (is inhabited)"
- parametric judgment: $\Pi x{:}A.B$ — "$B[s/x]$ (is inhabited) for all $s : A$"

References: Curry-Howard for DTT


```
@InCollection{martinlof,
  author =         "P. Martin-{L\"o}f",
  booktitle =      "Proceedings of the '73 Logic
      Colloquium",
  title =          "An Intuitionistic Theory of Types:
      Predicative Part",
  publisher =      "North-Holland",
  year =           "1974",
}

@Article{lcccseely,
  author =         "R. Seely",
  title =          "Locally cartesian closed categories
      and type theory",
  journal =        "Math. Proc. Cambridge Philos. Soc.",
  volume =         "95",
  pages =          "33--48",
  year =           "1984",
}
```

References: judgments as types


```
@Article{martinlof_meaning,
  title =          "On the Meanings of the Logical
     Constants and the Justifications of the Logical
     Laws",
  author =         "P. Martin-L{\"o}f",
  journal =        "Nordic Journal of Philosophical Logic
     ",
  volume =         "1",
  number =         "1",
  pages =          "3--10",
  year =           "1996"
}

@InCollection{logicalframeworks,
 author = {F. Pfenning},
 title = {Logical frameworks},
 booktitle = {Handbook of automated reasoning},
 publisher = {Elsevier},
 year = {2001},
 pages = {1063--1147}
}
```