Type Theory Study Group

Meeting 1.5

Special Topics with:

Jon Sterling

Host:

Joseph Abrahamson

Guest:

Darin Morrison

28.11.2015

Notes by:

Ilan Godik

(1.5) (1) Special Topics with Jon Sterling
Host: Joseph Abrahamson.
with: Darin Morrison

ABT's

Specifying the syntax,
we haven't gotten into the semantics just yet.

ABTs: Syntax with binding.
* appreciate the tools we'll use later.
"Simply implement this type theory", but then all this binding mess.

There are different binding frameworks

- PFPL makes one choice
- Many trade-offs, power, usability & others.

1.5 2

ASTs & ABTs

ASTs: Abstract Syntax Trees
Describe the structure as a tree, quite raw.
Using variables in math seems very intuitive,
but formally specifying them & implementing
them isn't so simple.

Darin: Higher Dimentional Logics & Category theory.
Jon: Came from linguistics, collaberating with Daring,
Jon Prl, Type refinements.
≠ Refinement types.
Very overloaded..

Sorts

Syntactic categories;
Like: - verbs, sentenses
- Expressions, String Literals, ...

ABTs as a Type Theory

- Sorts as types
- Inhabitants as elements of the types.

* Call them sorts not to confuse with types
we work with inside our type theory.

Arities/Valence: a seq. of sorts

1.5 3

Logical Framework vs. Type Theory

As per Martin Löf, the logical framwork deals with the matters of syntax, bootstrapping our type theory.

* Many Logical Frameworks are Type Theories but not the other way around.
* We want the Logical Framework to be decidable, and not nessesarily so with type theory.

The types of semantics we give for them are different.

Need to be careful not to work only via the syntax, We may get to a meaningless type theory without semantics backing us up.

* See Frank Pfenning's Lectures & Lecture notes.
* Don't $\overset{\text{gain}}{\text{or lose}}$ information, symmetry between Introduction & Elimination.
(Logical Harmony)

Verificationism:

Specify everything by Introductions, * How something comes to be introducing normal forms.

Addmissibility: Define elims. by what can you do with the info. of these intros.

Pragmatism:

* How you can use something

Specify everything by Elimination, and get the introduction rules that fit them.

* Useful to be not committing to either side.

1.5 5

ABT implementations

Used for implementing languages.

Jon implemented several ABT libs.

- 1 Haskell lib - "A wierd one"
- 2 SML libs
- 1 Purescript lib

* Seperate the local repr.
  - Locally nameless with De. Brujin Indices
  - Views from the outside with names.

* Jon warns against using the Haskell one.
- An experiment to make only well formed at compile time, HLists and such.
- We check well formedness anyway at runtime, use existentials, GADTs pattern matches...
- Performance issues.
- Not worth it at the long run.
- Unisorted.
- Nice for understanding maybe.

(1.5) (6)

De. Brujin Indices

- Simpler
- Performance issues, can deal with.
- Better expose a nice user interface on top of de Brujin indices
- Nice for deductive purposes, go implement one yourself!
- One form for α-equivalent terms.

Jon's paper: Syntax & Semantics of ABTs

- Clear some points
- Extend for symbolic parameters.
- Symbols different from variables, subject only to injective mapping, injective - 2 different symbols can't go to the same symbols, no substitution.

- Exeptions, mutable references, open sums.
- Sheafs on collections of symbols

1.5 7

[Pre]Sheafs? At a high level

- Very abstract
- Example: ...?
- Lots of smart words :)
- Express gnarly stuff neatly
- Consistant views that can be glued together such that they still represent the whole
- Continious Truth.
  Choice sequences.
  Finite views, which aren't conflicting, retain some sort of information.

Books

* Category Theory
  - ✓ Taylor's Practical Foundation of Mathematics great examples from computer science.
  - Maclane's book - thorough.
  - Awodey's book a bit more beginner friendly.
* Topology via Logic