Syntax and Semantics of Quantitative Type Theory

Robert Atkey
Strathclyde University, Glasgow
robert.atkey@strath.ac.uk

LICS 2018
11th July 2018

Dependency and Accountancy

In Martin-Löf Type Theory

$$x_1 : S_1, \ldots, x_n : S_n \vdash M : T$$

In Martin-Löf Type Theory

$$x_1 : S_1, \ldots, x_n : S_n \vdash M : T$$

variables $x_1, \ldots, x_n$ are mixed usage

$$n : Nat, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

$$n : \mathit{Nat}, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

x is used *computationally*

$$n : Nat, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

x is used *computationally*

n is used *logically*

In Linear Logic

$$x_1 : X_1, \ldots, x_n : X_n \vdash M : Y$$

In Linear Logic

$$x_1 : X_1, \ldots, x_n : X_n \vdash M : Y$$

presence of a variable records usage
each x_i must be used by M exactly once

$$n : \mathsf{Nat}, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

Can we read this judgement linearly?

$$n : \mathsf{Nat}, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

Can we read this judgement linearly?

▷ n appears in the context, but is not used computationally

$$n : \mathsf{Nat}, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

Can we read this judgement linearly?

- n appears in the context, but is not used computationally
- n appears *twice* in types

$$n : \mathsf{Nat}, x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

Can we read this judgement linearly?

- n appears in the context, but is not used computationally
- n appears *twice* in types

Is n even used at all?

$$n : \mathsf{Nat} \mid x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

$$n : \mathsf{Nat} \mid x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

- Separate *intuitionistic* / *unrestricted* uses and *linear* uses

$$n : \mathsf{Nat} \mid x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

- Separate *intuitionistic* / *unrestricted* uses and *linear* uses
- Types can depend on intuitionistic data, but not linear data

will come back to this...

$$n : \mathsf{Nat} \mid x : \mathsf{Fin}(n) \vdash x : \mathsf{Fin}(n)$$

- Separate *intuitionistic* / *unrestricted* uses and *linear* uses
- Types can depend on intuitionistic data, but not linear data

will come back to this...

(Barber, 1996)
(Cervesato and Pfenning, 2002)
(Krishnaswami, Pradic, and Benton, 2015)
(Vákár, 2015)

Separation interferes with dependency:

$$n : \mathsf{Nat} \mid x : \mathsf{Fin}(n) \vdash (x, \mathsf{refl}(x)) : (y : \mathsf{Fin}(n)) \times (x \equiv y)$$

Separation interferes with dependency:

$$n : \mathsf{Nat} \mid x : \mathsf{Fin}(n) \vdash (x, \mathsf{refl}(x)) : (y : \mathsf{Fin}(n)) \times (x \equiv y)$$

$$n : \mathsf{Nat}, x : \mathsf{Fin}(n) \mid \hat{x} : \widehat{\mathsf{Fin}}(n, x) \vdash (x, \hat{x}, \mathsf{refl}(x)) : (y : \mathsf{Fin}(n)) \times \widehat{\mathsf{Fin}}(n, y) \otimes (x \equiv y)$$

Quantitative Coeffect calculi:

$$x_1 \overset{\rho_1}{:} S_1, \ldots, x_n \overset{\rho_n}{:} S_n \vdash M : T$$

Quantitative Coeffect calculi:

$$x_1 \overset{\rho_1}{:} S_1, \ldots, x_n \overset{\rho_n}{:} S_n \vdash M : T$$

- ▷ The ρ_i record usage from some semiring R
  - . $1 \in R$ — a use
  - . $0 \in R$ — not used
  - . $\rho_1 + \rho_2$ — adding up uses (e.g., in an application)
  - . $\rho_1 \rho_2$ — nested uses

Quantitative Coeffect calculi:

$$x_1 \overset{\rho_1}{:} S_1, \ldots, x_n \overset{\rho_n}{:} S_n \vdash M : T$$

- The ρ_i record usage from some semiring R
  - $1 \in R$ — a use
  - $0 \in R$ — not used
  - $\rho_1 + \rho_2$ — adding up uses (e.g., in an application)
  - $\rho_1 \rho_2$ — nested uses

(Petricek, Orchard, and Mycroft, 2014)
(Brunel, Gaboardi, Mazza, and Zdancewic, 2014)
(Ghica and Smith, 2014)

Can we adapt this idea to dependent types?

Can we adapt this idea to dependent types?

McBride's idea:

- allow 0-usage data to appear in types.
  (McBride, 2016)

Can we adapt this idea to dependent types?

McBride's idea:

- ▷ allow 0-usage data to appear in types.
  (McBride, 2016)

$$x_1 \overset{\rho_1}{:} S_1, \ldots, x_n \overset{\rho_n}{:} S_n \vdash M \overset{\sigma}{:} T$$

where $\sigma \in \{0, 1\}$.

- ▷ $\sigma = 1$ — the "real" computational world
- ▷ $\sigma = 0$ — the types world

(allowing arbitrary ρ yields a system where substitution is inadmissible)

Can we adapt this idea to dependent types?

McBride's idea:

- allow 0-usage data to appear in types.
  (McBride, 2016)

$$x_1 \overset{\rho_1}{:} S_1, \ldots, x_n \overset{\rho_n}{:} S_n \vdash M \overset{\sigma}{:} T$$

where $\sigma \in \{0, 1\}$.

- $\sigma = 1$ — the "real" computational world
- $\sigma = 0$ — the types world

(allowing arbitrary ρ yields a system where substitution is inadmissible)

Zero-ing is an admissible rule: $\dfrac{\Gamma \vdash M \overset{1}{:} T}{0\Gamma \vdash M \overset{0}{:} T}$ allowing promotion to the type world.

Quantitative Type Theory

Contexts

$$\frac{}{\diamond \vdash}\ \textsc{Emp} \qquad\qquad \frac{\Gamma \vdash \qquad 0\Gamma \vdash S}{\Gamma, x \overset{\rho}{:} S \vdash}\ \textsc{Ext}$$

Contexts

$$\frac{}{\diamond \vdash}\ \textsc{Emp} \qquad \frac{\Gamma \vdash \qquad 0\Gamma \vdash S}{\Gamma, x \overset{\rho}{:} S \vdash}\ \textsc{Ext}$$

Types

$$0\Gamma \vdash S$$

Contexts

$$\frac{}{\diamond \vdash}\ \textsc{Emp} \qquad \frac{\Gamma \vdash \qquad 0\Gamma \vdash S}{\Gamma, x \overset{\rho}{:} S \vdash}\ \textsc{Ext}$$

Types

$$0\Gamma \vdash S$$

Terms

$$\frac{0\Gamma, x \overset{\sigma}{:} S, 0\Gamma' \vdash}{0\Gamma, x \overset{\sigma}{:} S, 0\Gamma' \vdash x \overset{\sigma}{:} S}\ \textsc{Var} \qquad \frac{\Gamma \vdash M \overset{\sigma}{:} S \qquad 0\Gamma \vdash S \equiv T}{\Gamma \vdash M \overset{\sigma}{:} T}\ \textsc{Conv}$$

Π-*type formation*

$$\frac{0\Gamma \vdash S \qquad 0\Gamma, x \overset{0}{:} S \vdash T}{0\Gamma \vdash (x \overset{\pi}{:} S) \to T}$$

Π-type formation

$$\frac{0\Gamma \vdash S \qquad 0\Gamma, x \overset{0}{:} S \vdash T}{0\Gamma \vdash (x \overset{\pi}{:} S) \to T}$$

Π-type introduction and elimination

$$\frac{\Gamma, x \overset{\sigma\pi}{:} S \vdash M \overset{\sigma}{:} T}{\Gamma \vdash \lambda x \overset{\pi}{:} S.M^T \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T}$$

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \qquad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2 \qquad \sigma' = 0 \Leftrightarrow (\pi = 0 \vee \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathrm{App}_{(x \overset{\pi}{:} S)T}(M, N) \overset{\sigma}{:} T[N/x]}$$

Π-*type elimination*

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \qquad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2 \qquad \sigma' = 0 \Leftrightarrow (\pi = 0 \vee \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathrm{App}_{(x \overset{\pi}{:} S) T}(M, N) \overset{\sigma}{:} T[N/x]}$$

There are three cases:

Π-*type elimination*

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \qquad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2 \qquad \sigma' = 0 \Leftrightarrow (\pi = 0 \lor \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathrm{App}_{(x \overset{\pi}{:} S)T}(M, N) \overset{\sigma}{:} T[N/x]}$$

There are three cases:

1. $\sigma = 1, \pi \neq 0, \sigma' = 1$

   Function uses its argument "computationally";

Π-*type elimination*

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \qquad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2 \qquad \sigma' = 0 \Leftrightarrow (\pi = 0 \vee \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathrm{App}_{(x \overset{\pi}{:} S)T}(M, N) \overset{\sigma}{:} T[N/x]}$$

There are three cases:

1. $\sigma = 1, \pi \neq 0, \sigma' = 1$
   Function uses its argument "computationally";
2. $\sigma = 1, \pi = 0, \sigma' = 0$
   Function yields computational data, but doesn't use its argument;

Π-*type elimination*

$$\frac{\Gamma_1 \vdash M \overset{\sigma}{:} (x \overset{\pi}{:} S) \to T \qquad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2 \qquad \sigma' = 0 \Leftrightarrow (\pi = 0 \vee \sigma = 0)}{\Gamma_1 + \pi\Gamma_2 \vdash \mathrm{App}_{(x \overset{\pi}{:} S)T}(M, N) \overset{\sigma}{:} T[N/x]}$$

There are three cases:

1. $\sigma = 1, \pi \neq 0, \sigma' = 1$
   Function uses its argument "computationally";
2. $\sigma = 1, \pi = 0, \sigma' = 0$
   Function yields computational data, but doesn't use its argument;
3. $\sigma = 0, \sigma' = 0$
   We are not using the function for its computational content, so π does not matter.

In the paper... additional type formers

1. Quantitative Σ-types;
2. Boolean type;
3. Universe of small sets (only exists in the $\sigma = 0$ fragment).

Zero-ing

$$\frac{\Gamma \vdash M \overset{\sigma}{:} S}{0\Gamma \vdash M \overset{0}{:} S}\ \textsc{Tm-Zero}$$

Zero-ing

$$\frac{\Gamma \vdash M \overset{\sigma}{:} S}{0\Gamma \vdash M \overset{0}{:} S}\ \textsc{Tm-Zero}$$

Zero needs nothing: $\qquad \Gamma \vdash M \overset{0}{:} S \qquad \Rightarrow \qquad 0\Gamma = \Gamma$

Zero-ing

$$\frac{\Gamma \vdash M \overset{\sigma}{:} S}{0\Gamma \vdash M \overset{0}{:} S}\ \textsc{Tm-Zero}$$

Zero needs nothing: $\quad \Gamma \vdash M \overset{0}{:} S \quad \Rightarrow \quad 0\Gamma = \Gamma$

Weakening

$$\frac{\Gamma, \Gamma' \vdash \mathcal{J} \qquad 0\Gamma \vdash U}{\Gamma, x \overset{0}{:} U, \Gamma' \vdash \mathcal{J}}\ \textsc{Weaken}$$

Zero-ing

$$\frac{\Gamma \vdash M \overset{\sigma}{:} S}{0\Gamma \vdash M \overset{0}{:} S}\ \textsc{Tm-Zero}$$

Zero needs nothing: $\Gamma \vdash M \overset{0}{:} S \quad \Rightarrow \quad 0\Gamma = \Gamma$

Weakening

$$\frac{\Gamma, \Gamma' \vdash \mathcal{J} \qquad 0\Gamma \vdash U}{\Gamma, x \overset{0}{:} U, \Gamma' \vdash \mathcal{J}}\ \textsc{Weaken}$$

Substitution

$$\frac{\Gamma_1, x \overset{\rho}{:} S, \Gamma' \vdash M \overset{\sigma}{:} T \qquad \Gamma_2 \vdash N \overset{\sigma'}{:} S \qquad 0\Gamma_1 = 0\Gamma_2 \qquad \sigma' = 0 \Leftrightarrow \rho = 0}{(\Gamma_1 + \rho\Gamma_2), \Gamma'[N/x] \vdash M[N/x] \overset{\sigma}{:} T[N/x]}$$

Semantic Interpretation

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;
2. $U : \mathcal{L} \to \mathcal{C}$ forgetting the computational content;

Quantitative Category with Families

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;
2. $U : \mathcal{L} \to C$ forgetting the computational content;
3. Addition and scaling structure on $\mathcal{L}$, fibred over U;

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;
2. $U : \mathcal{L} \to C$ forgetting the computational content;
3. Addition and scaling structure on $\mathcal{L}$, fibred over U;
4. Semantic types formed with respect to C:

$$S \in \mathrm{Ty}(\Delta), \qquad \Delta \in \mathrm{Ob}C;$$

Quantitative Category with Families

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;
2. $U : \mathcal{L} \to \mathcal{C}$ forgetting the computational content;
3. Addition and scaling structure on $\mathcal{L}$, fibred over U;
4. Semantic types formed with respect to $\mathcal{C}$:

$$S \in \mathrm{Ty}(\Delta), \qquad \Delta \in \mathrm{Ob}\,\mathcal{C};$$

5. Semantic terms, resourced and unresourced:

$$\begin{aligned} M &\in \mathrm{Tm}(\Delta, S), && \Delta \in \mathrm{Ob}\,\mathcal{C}, S \in \mathrm{Ty}(\Delta) \\ M &\in \mathrm{RTm}(\Gamma, S), && \Gamma \in \mathrm{Ob}\,\mathcal{L}, S \in \mathrm{Ty}(U\Gamma); \end{aligned}$$

Quantitative Category with Families

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;
2. $U : \mathcal{L} \to \mathcal{C}$ forgetting the computational content;
3. Addition and scaling structure on $\mathcal{L}$, fibred over U;
4. Semantic types formed with respect to $\mathcal{C}$:

$$S \in \mathrm{Ty}(\Delta), \qquad \Delta \in \mathrm{Ob}\mathcal{C};$$

5. Semantic terms, resourced and unresourced:

$$\begin{array}{ll} M \in \mathrm{Tm}(\Delta, S), & \Delta \in \mathrm{Ob}\mathcal{C}, S \in \mathrm{Ty}(\Delta) \\ M \in \mathrm{RTm}(\Gamma, S), & \Gamma \in \mathrm{Ob}\mathcal{L}, S \in \mathrm{Ty}(U\Gamma); \end{array}$$

6. Semantic zero-ing: $U : \mathrm{RTm}(\Gamma, S) \to \mathrm{Tm}(U\Gamma, S)$;

1. Category $\mathcal{L}$ for interpreting contexts and simultaneous substitutions;
2. $U : \mathcal{L} \to \mathcal{C}$ forgetting the computational content;
3. Addition and scaling structure on $\mathcal{L}$, fibred over U;
4. Semantic types formed with respect to $\mathcal{C}$:

$$S \in \mathrm{Ty}(\Delta), \qquad \Delta \in \mathrm{Ob}\mathcal{C};$$

5. Semantic terms, resourced and unresourced:

$$\begin{array}{ll} M \in \mathrm{Tm}(\Delta, S), & \Delta \in \mathrm{Ob}\mathcal{C}, S \in \mathrm{Ty}(\Delta) \\ M \in \mathrm{RTm}(\Gamma, S), & \Gamma \in \mathrm{Ob}\mathcal{L}, S \in \mathrm{Ty}(U\Gamma); \end{array}$$

6. Semantic zero-ing: $U : \mathrm{RTm}(\Gamma, S) \to \mathrm{Tm}(U\Gamma, S)$;
7. Resourced counterparts of substitution and comprehension, preserved by U.

R-Linear Combinatory Algebras (R-LCAs)
(Abramsky, Haghverdi, and Scott, 2002)

A carrier $\mathcal{A}$ with an application operator $(\cdot)$ and unary operators $!_\rho$, $\rho \in R$, and:

$$\begin{array}{lcl}
B \cdot x \cdot y \cdot z & = & x \cdot (y \cdot z) \\
C \cdot x \cdot y \cdot z & = & x \cdot z \cdot y \\
I \cdot x & = & x \\
K \cdot x \cdot !_0 y & = & x \\
W_{\pi\rho} \cdot x \cdot !_{\pi+\rho} y & = & x \cdot !_\pi y \cdot !_\rho y \\
D \cdot !_1 x & = & x \\
\delta_{\pi\rho} \cdot !_{\pi\rho} x & = & !_\pi !_\rho x \\
F_\rho \cdot !_\rho x \cdot !_\rho y & = & !_\rho (x \cdot y)
\end{array}$$

R-Linear Combinatory Algebras (R-LCAs)
(Abramsky, Haghverdi, and Scott, 2002)

A carrier $\mathcal{A}$ with an application operator $(\cdot)$ and unary operators $!_\rho$, $\rho \in R$, and:

$$
\begin{array}{lcl}
B \cdot x \cdot y \cdot z & = & x \cdot (y \cdot z) \\
C \cdot x \cdot y \cdot z & = & x \cdot z \cdot y \\
I \cdot x & = & x \\
K \cdot x \cdot !_0 y & = & x \\
W_{\pi\rho} \cdot x \cdot !_{\pi+\rho} y & = & x \cdot !_\pi y \cdot !_\rho y \\
D \cdot !_1 x & = & x \\
\delta_{\pi\rho} \cdot !_{\pi\rho} x & = & !_\pi !_\rho x \\
F_\rho \cdot !_\rho x \cdot !_\rho y & = & !_\rho (x \cdot y)
\end{array}
$$

For example: any LCA gives a $0, 1, \omega$-LCA; any BCI algebra gives a $\mathbb{N}$-LCA; any SK algebra gives a $\{0, 1\}$-LCA.

1. Let $\mathcal{C} = \mathrm{Set}$, category of sets and functions
2. Let $\mathcal{L}$ be Assemblies over $\mathcal{A}$

   sets with computational information
3. $U : \mathcal{L} \to \mathcal{C}$ forgets the computational information
4. Types $S \in \mathrm{Ty}(\Delta)$ include computational information, but:

   only depend on non-computational part
5. Terms $M \in \mathrm{RTm}(\Gamma, S)$ are tracked by realisers from $\mathcal{A}$
6. Terms $M \in \mathrm{Tm}(\Delta, S)$ are set theoretic functions

Read constructively, yields an “efficient” compilation method for QTT, which respects and uses the usage information.

Main contribution

An interpretation of Type Theory that allows for “sub-computational” models

Main contribution

An interpretation of Type Theory that allows for "sub-computational" models

1. No longer need the full power of SK;
2. *R*-LCAs allow for interesting refined models of computation;
3. For example — *(future work!)*
   3.1 Hofmann and Dal Lago's realisability models of Implicit Computational Complexity;
   3.2 Geometry of Interaction models,
   incl Abramsky's *A structural approach to reversible computation, 2005*;
   3.3 Imperative models, after Ahmed, Fluet, Morrisett's L^3;
   3.4 Staged models

Summary

- Quantitative Type Theory:
  Fixed and extended formulation of McBride's "Plenty o' Nuttin"' system

- Categorical and Realisability models
  QCwFs, and models in *R*-LCAs

▷ Quantitative Type Theory:
  Fixed and extended formulation of McBride's "Plenty o' Nuttin"' system

▷ Categorical and Realisability models
  QCwFs, and models in R-LCAs

Future work

- More R-LCAs, for more applications (ICC, reversible computation, ...)
- Combination with split context systems
- Implementation (already started in Idris)
- Internalisation of extensional properties yielded by quantitative information

- ▷ Quantitative Type Theory:
  Fixed and extended formulation of McBride's "Plenty o' Nuttin"' system

- ▷ Categorical and Realisability models
  QCwFs, and models in R-LCAs

Future work

- More R-LCAs, for more applications (ICC, reversible computation, ...)
- Combination with split context systems
- Implementation (already started in Idris)
- Internalisation of extensional properties yielded by quantitative information

Thank You for Listening!