On the Syntax and Semantics of Quantitative Typing

Andreas Abel[1]

[1]Department of Computer Science and Engineering
Chalmers and Gothenburg University, Sweden

Workshop on Mixed Inductive-Coinductive Reasoning
19 April 2018

Introduction

- Quantitative typing generalizes linear typing.
- Practical uses:
  - Cardinality analysis in compilers: strictness, dead code.
  - Differential privacy.
  - Erasure in type theory (EPTS, Idris).
  - Security typing!?
- Theory: graded comonads.
- Thesis:

  The generalization of linear typing to quantitative typing allows a smooth integration with dependent typing.

A Free Theorem from linear typing

Theorem (Bob Atkey)

Given an abstract type K of "keys" with operation

$$\mathsf{compare} : (K \otimes K) \multimap (\mathsf{Bool} \otimes K \otimes K)$$

and a program (i.e., closed term)

$$f : \mathsf{List}\, K \multimap \mathsf{List}\, K$$

then f is a list permutation.

Proof formalized in Agda.

https://github.com/bobatkey/sorting-types. □

Proof of the free theorem

- Category $\mathbb{W}$ of lists over K and permutations $\bowtie$.
- $\mathbb{W}$ is symmetric monoidal: $\mathbb{1}$ = empty list, $\otimes$ is concatenation.
- Logical relation $\models_A \subseteq \mathbb{W} \times A$ natural in $\mathbb{W}$ (i.e., closed under $\bowtie$).
- $w \models_A a$: *value a can be constructed exactly from the resources w.*

$$
\begin{array}{lcl}
w \models_1 () & \iff & w = \mathbb{1} \\
w \models_{A_1 \oplus A_2} \mathsf{in}_i(a) & \iff & w \models_{A_i} a \\
w \models_{A \otimes B} (a, b) & \iff & w \bowtie w_1 \otimes w_2 \text{ and } w_1 \models_A a \text{ and } w_2 \models_B b \\
 & & \quad \text{for some } w_1, w_2 \\
w \models_{A \multimap B} f & \iff & w' \models_A a \text{ implies } w \otimes w' \models_B f(a) \text{ for all } w'
\end{array}
$$

- Setting: $w \models_K k$ iff w is singleton k.
- Remember: $\mathsf{List}\,K = 1 \oplus (K \otimes \mathsf{List}\,K)$.
- Consequence: $w \models_{\mathsf{List}\,K} ks$ iff w is a permutation of ks.

Proof of the free theorem (ctd.)

- Fundamental theorem: If $\Gamma \vdash t : A$ and $w \models_\Gamma \sigma$ then $w \models_A t\sigma$.
- $\vdash f : \mathsf{List}\ K \multimap \mathsf{List}\ K$ implies $\mathbb{1} \models_{\mathsf{List}\ K \multimap \mathsf{List}\ K} f$
- With $ks \models_{\mathsf{List}\ K}\ ks$ have $\mathbb{1} \otimes ks \models f(ks)$, thus $ks \bowtie f(ks)$.

Remarks:

- We call the world w of (mandatorily) consumable resources *support*.
- Elements of *closed* types (not mentioning K) have *empty* support.
- Eliminators like $\mathsf{if} : \mathsf{Bool} \multimap (A \& A) \multimap A$ use additive conjunction $\&$.

$$w \models_{A \& B} (a, b) \iff w \models_A a \text{ and } w \models_B b$$

- Subexponentials for $n \in \mathbb{N}$ where $w^n = w \otimes \ldots \otimes w$ (n times):

$$\begin{array}{lcl} w \models_{!^n A} a & \iff & w \bowtie w'^n \text{ and } w' \models_A a \\ w \models_{?^n A} a & \iff & w^n \models_A a \end{array}$$

- Gives quadratic functions like $\lambda^2 x.\,(x, x) : !^2 A \multimap A \times A$. But affine?

Choice of resources

- Given an abstract type K with $e : K$ and $\_ \cdot \_ : K \multimap K \multimap K$ and a boolean $b : B$ consider

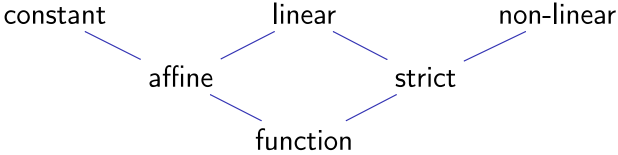
$$\begin{array}{lcl} \lambda^{\{0,1\}}x.\ \text{if}\ b\ \text{then}\ x\ \text{else}\ e & : & !^{\{0,1\}}K \multimap K \\ \lambda^{\{1,2\}}x.\ \text{if}\ b\ \text{then}\ x\ \text{else}\ x \cdot x & : & !^{\{1,2\}}K \multimap K \end{array}$$

  There is imprecision in the quantity of usage of x.

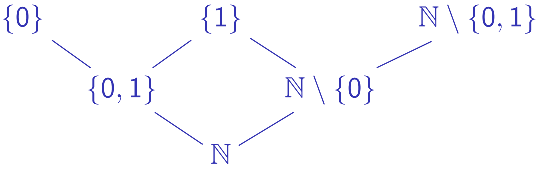
- In general, we want $!^q A \multimap B$ for $q \subseteq \mathbb{N}$.
- We extend $\mathbb{W}$ by non-empty additive products $\&_{i \in q} A_i$ (infima).
- Morphisms $w \leq w'$ include dropping of alternatives $A \,\&\, B \leq A$, in general, $\&_{i \in q} A_i \leq \&_{j \in q'} A_j$ for $q' \subseteq q$.
- Exponent: $w^q = \&_{n \in q} w^n$.
- $w \models_{!^q A} a$ iff $w' \models_A a$ for some w' with $w \leq w'^q$.
- Uninformed function type $A \to B$ is $!^{\mathbb{N}} A \multimap B$.

Quantity lattice

- Function classification:



- Expressed as quantitative information $q \subseteq \mathbb{N}$ in $(!^q A) \multimap B$:



- $\mathsf{Q} = \{\{0\}, \{1\}, \leq 1, \geq 2, \geq 1, \mathbb{N}\}$

Function composition

- Multiplication $q \cdot r = \{m \cdot n \mid m \in q, n \in r\}$ rounded up to be in Q.

$q \cdot r$	$\{0\}$	$\{1\}$	≤ 1	≥ 2	≥ 1	$\mathbb{N}$
$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$	$\{0\}$
$\{1\}$	$\{0\}$	$\{1\}$	≤ 1	≥ 2	≥ 1	$\mathbb{N}$
≤ 1	$\{0\}$	≤ 1	≤ 1	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$
≥ 2	$\{0\}$	≥ 2	$\mathbb{N}$	≥ 2	≥ 2	$\mathbb{N}$
≥ 1	$\{0\}$	≥ 1	$\mathbb{N}$	≥ 2	≥ 1	$\mathbb{N}$
$\mathbb{N}$	$\{0\}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$	$\mathbb{N}$

- Addition $q + r = \{m + n \mid m \in q, n \in r\}$ rounded up to be in Q.
- Addition for summing usage quantities in two terms.

Dependent linear types

- Multiplicative linear dependent function and pair types.

$$
\begin{array}{lcl}
w \models_{\Pi\, A\, F} f & \iff & w' \models_A a \text{ implies } w \otimes w' \models_{F(a)} f(a) \text{ for all } w' \\
w \models_{\Sigma\, A\, F} (a, b) & \iff & w_1 \models_A a \text{ and } w_2 \models_{F(a)} b \text{ for some } w_1, w_2 \\
 & & \text{with } w \leq w_1 \otimes w_2
\end{array}
$$

- Obvious, no?

Dependent linear types, what took you so long?

- 1972: Martin-Löf: (Dependent) Type Theory
- 1987: Girard: Linear logic
- *(3 decades later)*
- 2016: McBride: I got plenty of nuttin'
- 2018: Atkey: Syntax and Semantics of Quantitative Type Theory
- What took us so long?
- (Wrong) paradigm:
  - Focus on structural rules (weakening, contraction).
  - Separate contexts for linear and intuitionistic assumptions.
  - Same quantity context for term and types.

$\Gamma \vdash t : A$ implies $\Gamma \vdash A : \mathsf{Type}$

Quantitative type theory

- Syntax ($q, r \in \mathsf{Q}$):

t, u, A, F	$::=$	x	name (free variable)
	$\mid$	$\lambda^q x.\, t$	λ-abstraction (binder) with quantity
	$\mid$	$t \cdot^q u$	application with quantity
	$\mid$	$\Pi^{q,r} A\, F$	dependent function type (no binder)
	$\mid$	U_ℓ	sort

- Usage calculation $\boxed{|t| : \mathsf{Var} \to \mathsf{Q}}$.

$$
\begin{array}{rcl}
|x| & = & x{:}1 \\
|t \cdot^q u| & = & |t| + q|u| \\
|\lambda^q x.\, t| & = & |t| \setminus x \\
|\mathsf{U}_\ell| & = & \emptyset \\
|\Pi^{q,r} A\, F| & = & |A| + |F|
\end{array}
$$

Quantitative typing

$$\frac{\vdash \Gamma}{\Gamma \vdash x : \Gamma(x)} \qquad \frac{\Gamma \vdash t : \Pi^{q,r} A\, F \qquad \Gamma \vdash u : A}{\Gamma \vdash t \cdot^{q} u : F \cdot^{r} u}$$

$$\frac{\Gamma, x{:}A \vdash t : F \cdot^{r} x}{\Gamma \vdash \lambda^{q} x.\, t : \Pi^{q,r} A\, F}\; q \supseteq |t|^{x}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathsf{U}_{\ell} : \mathsf{U}_{\ell'}}\; \ell < \ell' \qquad \frac{\Gamma \vdash A : \mathsf{U}_{\ell} \qquad \Gamma \vdash F : A \xrightarrow{r} \mathsf{U}_{\ell}}{\Gamma \vdash \Pi^{q,r} A\, F : \mathsf{U}_{\ell}}$$

$$\frac{\Gamma \vdash t : A \qquad \Gamma \vdash A \leq B}{\Gamma \vdash t : B}$$

Subtyping

$$\frac{\Gamma \vdash A = A' : \mathsf{U}_\ell}{\Gamma \vdash A \leq A'}$$

$$\frac{\vdash \Gamma}{\Gamma \vdash \mathsf{U}_\ell \leq \mathsf{U}_{\ell'}} \; \ell \leq \ell'$$

$$\frac{\Gamma \vdash A' \leq A \qquad \Gamma, x{:}A' \vdash F \cdot^r x \leq F' \cdot^r x}{\Gamma \vdash \Pi^{q,r} A\, F \leq \Pi^{q',r} A'\, F'} \; q \subseteq q'$$

Related Work

- Simple types: abundance of quantitative type systems.
- McBride 2016: $Q = \{\{0\}, \{1\}, \mathbb{N}\}$. Usage in types does not count!
- Atkey 2018, QTT: Q semiring.
- Brady: implementing McBride/Atkey system in Idris 2.

Future work

- CwF-like model for my variant of QTT.
- Internalize free theorems from linearity?!
- Relate to other modal type theories.
- Add to Agda.