Type Theory and Implicit Computational Complexity

Robert Atkey

Lecture 1

30th July 2024

SPLV 2024

Tuesday : Linear Types and PTIME

Wednesday : Soundness via Realisability

Thursday : Linear Dependent Types

Friday : QTT + ICC

$$A, B ::= I \mid A \otimes B \mid A \multimap B \mid a$$

affine linear λ-calculus

$$\frac{}{\Gamma, x : A \vdash x : A}$$

$\Gamma \vdash M : A \;\sim\; \Gamma, \Delta \vdash M : A$

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x . M : A \multimap B} \qquad \frac{\Gamma \vdash M : A \multimap B \quad \Delta \vdash N : A}{\Gamma, \Delta \vdash M N : B}$$

$$\frac{\Gamma \vdash M : A \quad \Delta \vdash N : B}{\Gamma, \Delta \vdash (M, N) : A \otimes B} \qquad \frac{\Gamma \vdash M : A \otimes B \quad \Delta, x : A, y : B \vdash N : C}{\Gamma, \Delta \vdash \text{let } (x, y) = M \text{ in } N : C}$$

$$\frac{}{\Gamma \vdash * : I} \qquad \frac{\Gamma \vdash M : I \quad \Delta \vdash N : C}{\Gamma, \Delta \vdash \text{let } * = M \text{ in } N : C}$$

Quantitative type systems — (Granule system)

$$A, B ::= I \mid A \otimes B \mid A \multimap B \mid a \mid !_\rho A$$

$$\Gamma ::= \varepsilon \mid \Gamma, x :^\rho A$$

$\Gamma_1 + \Gamma_2$ $\quad \varepsilon + \varepsilon = \varepsilon$

$$(\Gamma_1, x :^{\rho_1} A) + (\Gamma_2, x :^{\rho_2} A) = (\Gamma_1 + \Gamma_2), x :^{\rho_1 + \rho_2} A$$

$$\frac{}{0\Gamma, x :^1 A, 0\Gamma' \vdash x : A}$$

$$\frac{\Gamma, x :^1 A \vdash M : B}{\Gamma \vdash \lambda x. M : A \multimap B} \qquad \frac{\Gamma_1 \vdash M : A \multimap B \quad \Gamma_2 \vdash N : A}{\Gamma_1 + \Gamma_2 \vdash M N : B}$$

exercise : $\otimes$, I rules

$$\frac{\Gamma \vdash M : A}{\rho\Gamma \vdash !_\rho M : !_\rho A} \qquad \frac{\Gamma_1 \vdash M : !_\rho A \quad \Gamma_2, x :^\rho A \vdash N : B}{\Gamma_1 + \Gamma_2 \vdash \mathbf{let}\ !_\rho x = M \text{ in } N : B}$$

Implicit Complexity via Linearity

- $(\lambda x. M) N \longrightarrow M[x := N]$

  size = size + 1
- Add $\forall$ $\quad \forall \alpha. \alpha \multimap \alpha \multimap \alpha \simeq \text{Bool}$

  $\forall \alpha. !(\alpha \multimap \alpha) \multimap (\alpha \multimap \alpha) \overset{?}{\simeq} \text{Nat}$

  $!A \multimap !A \otimes !A$ full linear logic

  $!_{p+q} A \multimap !_p A \otimes !_q A$ Bounded Linear Logic

  $!A \multimap A^n$ Soft linear logic

Why not just add Datatypes?

$$\frac{}{\vdash ze : Nat} \qquad \frac{\Gamma \vdash N : Nat}{\Gamma \vdash su\ N : Nat} \qquad \frac{\vdash M_z : A \quad x : A \vdash M_s : A \quad \Gamma \vdash N : Nat}{\Gamma \vdash iter(M_z, x.\ M_s, N) : A}$$

$dup : Nat \multimap Nat \otimes Nat$

$dup = iter((ze, ze), (m, n).(su\ m, su\ n))$

$add : Nat \multimap Nat \multimap Nat$

$add = iter(\lambda y. y, r. \lambda y. su(r\ y))$

$0 + y = y$ $\qquad (1+x)+y = 1+(x+y)$

$mul : Nat \multimap Nat \multimap Nat$

$mul = iter(\lambda y. ze, r. \lambda y. let\ (y, y') = dup\ y\ in$
$\qquad add\ y\ (r\ y'))$

$exp : Nat \multimap Nat \multimap Nat$

$exp = iter(\lambda y. su\ ze, r. \lambda y. let\ (y, y') = dup\ y\ in$
$\qquad mul\ y\ (r\ y'))$

Polynomial Time is "Feasible"

- Iterate through the input
- Feasibility composes
- Iterations nest.

NIMBYish : ban construction (of useful things).