

Second-Order Generalised Algebraic Theories

Szumi Xie

Eötvös Loránd University, Budapest

SPLV 2024, Glasgow

What's the simplest programming language?

What's the simplest programming language? (algebraically)

Tm : Sort

$- \cdot -$: $Tm \rightarrow Tm \rightarrow Tm$

K : Tm

S : Tm

$K\beta$: $K \cdot x \cdot y = x$

$S\beta$: $S \cdot x \cdot y \cdot z = x \cdot z \cdot (y \cdot z)$

What's the simplest programming language? (SOGAT-ically)

Tm : Sort

$- \cdot -$: $Tm \rightarrow Tm \rightarrow Tm$

lam : $(Tm \rightarrow Tm) \rightarrow Tm$

β : $(lam\ t) \cdot u = t\ u$

What's the simplest programming language? (SOGAT-ically)

Tm : Sort

$-\cdot-$: $Tm \rightarrow Tm \rightarrow Tm$

lam : $(Tm \rightarrow Tm) \rightarrow Tm$

β : $(lam\ t) \cdot u = t\ u$

$$\frac{\Gamma \vdash t \quad \Gamma \vdash u}{\Gamma \vdash t \cdot u}$$

$$\frac{\Gamma, x \vdash t}{\Gamma \vdash lam\ t}$$

$$\frac{\Gamma, x \vdash t \quad \Gamma \vdash u}{\Gamma \vdash (lam\ t) \cdot u = t[u/x]}$$

First-order-ification

First-order-ification

Con : Sort

Sub : Con \rightarrow Con \rightarrow Sort

Tm : Con \rightarrow Sort

id : Sub Γ Γ

$- \circ -$: Sub Δ $\Gamma \rightarrow$ Sub Θ $\Delta \rightarrow$ Sub Θ Γ

$-[-]$: Tm $\Gamma \rightarrow$ Sub Δ $\Gamma \rightarrow$ Tm Δ

\diamond : Con

ε : Sub Γ \diamond

$- \triangleright$: Con \rightarrow Con

p : Sub $(\Gamma \triangleright) \Gamma$

q : Tm $(\Gamma \triangleright)$

$-, -$: Sub Δ $\Gamma \rightarrow$ Tm $\Delta \rightarrow$ Sub Δ $(\Gamma \triangleright)$

$-\cdot -$: Tm $\Gamma \rightarrow$ Tm $\Gamma \rightarrow$ Tm Γ

lam : Tm $(\Gamma \triangleright) \rightarrow$ Tm Γ

$\text{id} \circ \sigma = \sigma$

$\sigma \circ \text{id} = \sigma$

$(\sigma \circ \delta) \circ \nu = \sigma \circ (\delta \circ \nu)$

$t[\text{id}] = t$

$t[\sigma \circ \delta] = t[\sigma][\delta]$

$\varepsilon = (\sigma : \text{Sub } \Gamma \diamond)$

$p \circ (\sigma, t) = \sigma$

$q[\sigma, t] = t$

$(p \circ \sigma, q[\sigma]) = \sigma$

$(t \cdot u)[\sigma] = (t[\sigma]) \cdot (u[\sigma])$

$(\text{lam } t)[\sigma] = \text{lam } (t[\sigma \circ p, q])$

β : $(\text{lam } t) \cdot u = t[\text{id}, u]$

Languages which can be defined as SOGATs

Non-substructural languages

Non-substructural languages:

- Simply typed lambda calculus
- First-order logic
- System F
- Dependent type theory

Languages which can be defined as SOGATs

Non-substructural languages:

- Simply typed lambda calculus
- First-order logic
- System F
- Dependent type theory
- Theory of SOGAT signatures

Languages which can be defined as SOGATs

Non-substructural languages:

- Simply typed lambda calculus
- First-order logic
- System F
- Dependent type theory
- Theory of SOGAT signatures
- Higher observational type theory