

A type system for monotonicity

Michael Arntzenius

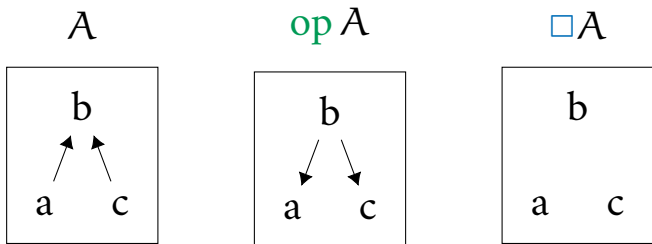
University of Birmingham

ICFP 2018

It's just the simply-typed λ -calculus!

$\llbracket A \rrbracket \in \mathbf{Poset}$

$\llbracket A \rightarrow B \rrbracket = \textit{monotone maps } \llbracket A \rrbracket \rightarrow \llbracket B \rrbracket,$
ordered pointwise



$f : \square A \rightarrow B$ is monotone iff

$$x = y \implies f(x) \leq f(y)$$

i.e. **always!**

setMap : $\square(\square A \rightarrow B) \rightarrow \text{Set } A \rightarrow \text{Set } B$

setMap f xs =

let box g = f in

do x ← xs

let box y = x

return (box (g (box y)))

$\text{setMap} : (\lambda A \rightarrow B) \rightarrow \text{Set } A \rightarrow \text{Set } B$

$\text{setMap } f \text{ } \chi s =$

do $x \leftarrow \chi s$

return $(f \text{ } x)$

$$A <: B$$
$$\text{id} : A \rightarrow B$$

$$[T]A <: B$$
$$\text{id} : TA \rightarrow B$$
$$T \in \{\text{id}, \text{op}, \square, \dots\}$$

$$f : TA \rightarrow B \quad g : UB \rightarrow C$$

$$g \circ f : (UT)A \rightarrow C$$

Monotonicity tames dragons!

1. Eventual consistency in distributed systems

<http://bloom-lang.net/calm/>

2. Determinism in parallel programs

LVars: Lattice-based Data Structures for Deterministic Parallelism,
Lindsey Kuper & Ryan Newton

3. Recursive queries in Datalog & Datafun

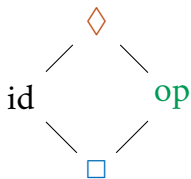
<http://www.rntz.net/datafun>

4. Paradoxes of self-reference

FIN

$\text{subtractEach} : \text{List } (\mathbb{Z} \times \text{op } \mathbb{Z}) \rightarrow \text{List } \mathbb{Z}$
 $\text{subtractEach } xs = [x - y \mid (x, y) \leftarrow xs]$

$$\begin{aligned}
 a \leq b : \text{id } A & \iff a \leq b : A \\
 a \leq b : \text{op } A & \iff a \geq b : A \\
 a \leq b : \square A & \iff a \leq b \wedge a \geq b : A \\
 a \leq b : \diamond A & \iff a \leq b \vee b \leq a : A
 \end{aligned}$$



		T			
		id	op	□	◇
U	id	id	op	□	◇
	op	op	id	□	◇
	□	□	□	□	◇
	◇	◇	◇	□	◇

SUBSUMPTION

$$\frac{\Gamma \vdash M : A \quad [T]A <: B}{T\Gamma \vdash M : C}$$