log(TESTING / PROGRAM)

favonia
U of Minnesota

Testing map

$$\mathrm{map} : \forall\alpha.\forall\beta.(\alpha\to\beta)\to\mathrm{list}(\alpha)\to\mathrm{list}(\beta)$$

Testing map

$$\text{map} : \forall\alpha.\forall\beta.(\alpha\to\beta)\to\text{list}(\alpha)\to\text{list}(\beta)$$

$$\alpha^* = \beta^* = \mathbb{N}$$

identity function and [0, 1, ..., n-1]

Testing exists

$$\text{exists} : \forall\alpha.(\alpha\to 2)\to \text{list}(\alpha)\to 2$$

$$\alpha^* = 2$$

identity function and all lists

Testing pick

$$\text{pick} : \forall\alpha.\ \alpha\times\alpha\rightarrow\alpha$$

$$\alpha^* = 2$$

$$(\text{true}, \text{false})$$

Theorem [Bernardy *et al.*]

$$p : \forall\alpha.\ (F(\alpha)\to\alpha)\times(G(\alpha)\to K)\to H(\alpha)$$

POS **POS** **POS**

$\alpha^* = \mu F$ *if it exists*

$\rho^* : F(\alpha)\to\alpha$ *catamorphism*
"roll"

Theorem [Bernardy *et al.*]

$$p : \forall \alpha.\ (F(\alpha) \to \alpha) \times (G(\alpha) \to K) \to H(\alpha)$$

POS **POS** **POS**

embed *project*

$$p : \forall \alpha.\ P(\alpha) \to H(\alpha)$$

POS

My Problems with Embedding/Projection

Type μF buried under layers of existence theorems.

Error-prone: *e.g.*, wrongly assuming $\text{list}(\alpha) \subseteq \mathbb{N} \times (\mathbb{N} \rightarrow \alpha)$.

$$p : \forall \alpha.\ P(\alpha) \rightarrow H(\alpha)$$

POS

Logarithmic Conjecture

$$p : \forall\alpha.\, P(\alpha) \rightarrow H(\alpha)$$

POS

$$\alpha^* = \mu\alpha.\log_\alpha P(\alpha)$$ *if it exists*

Logarithmic Conjecture

$$p : \forall\alpha.\, P(\alpha) \rightarrow H(\alpha)$$

POS

$$\alpha^* = \mu\alpha.\log_\alpha P(\alpha)$$ *if it exists*

Goal: simple and direct calculation

Testing pick

$$\text{pick} : \forall\alpha.\ \alpha\times\alpha\to\alpha$$

$$\alpha^* = \log_\alpha(\alpha\times\alpha) = \log_\alpha(\alpha^2) = 2$$

"all locations of α"
"all ways to generate an α-element"

$$\log_{\alpha}\alpha = 1 \qquad \log_{\alpha}K = 0$$

$$\log_\alpha \alpha = 1 \qquad \log_\alpha K = 0$$

$$\log_\alpha(A \times B) = \log_\alpha A + \log_\alpha B$$

$$\log_\alpha(A + B) = \log_\alpha A \cup \log_\alpha B$$

$A \cup B \approx$ "max(A,B)"

can be A+B, ideally more optimized

$\log_\alpha \alpha = 1$ $\log_\alpha K = 0$

$\log_\alpha (A \times B) = \log_\alpha A + \log_\alpha B$

$\log_\alpha (A + B) = \log_\alpha A \cup \log_\alpha B$

$A \cup B \approx$ "max(A,B)"

can be A+B, ideally more optimized

$\log_\alpha (A \cup B) = \log_\alpha A \cup \log_\alpha B$ *optional*

$$\log_\alpha \alpha = 1 \qquad \log_\alpha K = 0$$

$$\log_\alpha(A \times B) = \log_\alpha A + \log_\alpha B$$

$$\log_\alpha(A + B) = \log_\alpha A \cup \log_\alpha B$$

$A \cup B \approx$ "max(A,B)"

can be A+B, ideally more optimized

$$\log_\alpha(A \cup B) = \log_\alpha A \cup \log_\alpha B$$ *optional*

$$\log_\alpha(A \to B) = A \times \log_\alpha B$$

"B^A"

Testing map

$$\text{map} : \forall\alpha.\forall\beta.(\alpha\to\beta)\to\text{list}(\alpha)\to\text{list}(\beta)$$

Testing map

$$\text{map} : \forall\alpha.\forall\beta.(\alpha\to\beta)\to\text{list}(\alpha)\to\text{list}(\beta)$$

$$P(\alpha) = (\alpha\to\beta) \times \text{list}(\alpha)$$

Testing map

$$\text{map} : \forall\alpha.\forall\beta.(\alpha\rightarrow\beta)\rightarrow\text{list}(\alpha)\rightarrow\text{list}(\beta)$$

$$P(\alpha) = (\alpha\rightarrow\beta) \times \text{list}(\alpha)$$

$$\log_\alpha P(\alpha) = \log_\alpha(\alpha\rightarrow\beta) + \log_\alpha \text{list}(\alpha)$$

Testing map

$$\text{map} : \forall\alpha.\forall\beta.(\alpha\to\beta)\to\text{list}(\alpha)\to\text{list}(\beta)$$

$$P(\alpha) = (\alpha\to\beta) \times \text{list}(\alpha)$$

$$\log_\alpha P(\alpha) = \underline{\log_\alpha(\alpha\to\beta)} + \log_\alpha \text{list}(\alpha)$$

$$= \alpha \times \log_\alpha \beta$$

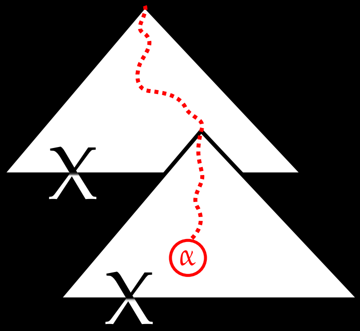
$$= \alpha \times 0$$

$$\cong 0$$

Rules of *Recursive* Logarithm

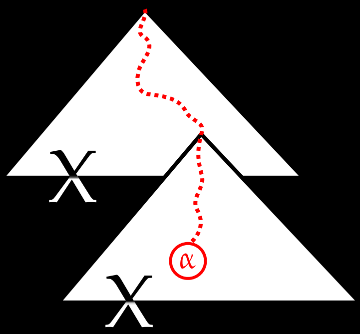
$$X \cong P(X)$$

$$\log_\alpha X \cong \log_\alpha P(X)$$



Rules of *Recursive* Logarithm

$$X \cong P(X)$$
$$\log_\alpha X \cong \log_\alpha P(X)$$



$$\log_\alpha(\mu X.P) = \mu X'.(\log_\alpha P)[\mu X.P/X]$$
$$\log_\alpha X = X'$$

Rules of *Recursive* Logarithm

$$\mathrm{list}(\alpha) = \mu X.\, 1 + \alpha \times X$$

$$\begin{aligned}\log_\alpha(\mathrm{list}(\alpha)) &= \mu X'.\, 0 \cup (1 + X') \\ &\approx \mu X'.\, 1 + X' \\ &= \mathbb{N}\end{aligned}$$

$$\log_\alpha(\mu X.P) = \mu X'.(\log_\alpha P)[\mu X.P/X]$$

$$\log_\alpha X = X'$$

I will assume $0 \cup A = A$ from now on

Testing map

$$\text{map} : \forall\alpha.\forall\beta.(\alpha\to\beta)\to\text{list}(\alpha)\to\text{list}(\beta)$$

$$P(\alpha) = (\alpha\to\beta) \times \text{list}(\alpha)$$

$$\log_\alpha P(\alpha) = \underbrace{\log_\alpha(\alpha\to\beta)}_{\cong\, 0} + \underbrace{\log_\alpha \text{list}(\alpha)}_{=\,\mathbb{N}}$$

Logarithmic Conjecture

$$p : \forall\alpha.\, P(\alpha) \rightarrow H(\alpha)$$

POS

$$\alpha^* = \mu\alpha.\log_\alpha P(\alpha)$$ *if it exists*

Goal: simple and direct calculation

Non-regular Recursion and Change of Bases / Chain Rules

$$N(\alpha) \cong 1 + \alpha \times N(\alpha{\times}\alpha)$$

$$\alpha - \alpha^2 - \alpha^4 - \alpha^8 - 1$$

example stolen from Categories of Containers [Abbott *et al.*]

Non-regular Recursion and Change of Bases / Chain Rules

$$N(\alpha) \cong 1 + \alpha \times N(\alpha \times \alpha)$$

$$\alpha - \alpha^2 - \alpha^4 - \alpha^8 - 1$$

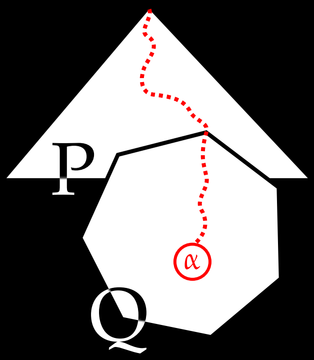
$$\log_\alpha N(\alpha) \cong 1 + \text{“}\log_{\alpha\times\alpha} N(\alpha\times\alpha)\text{”} \times \log_\alpha(\alpha\times\alpha)$$

$$N'(\alpha) \cong 1 + N'(\alpha\times\alpha) \times 2$$

example stolen from Categories of Containers [Abbott *et al.*]

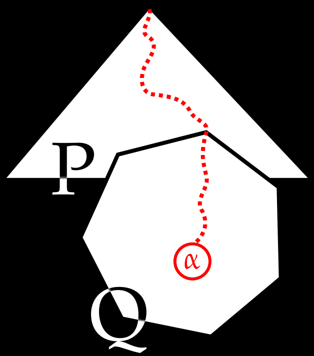
Non-regular Recursion and Change of Bases / Chain Rules

$$\log_{\alpha} P(Q(\alpha)) = P'(Q(\alpha)) \times Q'(\alpha)$$



Non-regular Recursion and Change of Bases / Chain Rules

$$\log_\alpha P(Q(\alpha)) = P'(Q(\alpha)) \times Q'(\alpha)$$



Should work for arbitrary P
with more equations

$$\log_\alpha((\alpha^2)^3) = 3 \times 2$$
$$\log_\alpha(\alpha^2 \times \alpha^2 \times \alpha^2) = 2 + 2 + 2$$

Non-regular Recursion and Change of Bases / Chain Rules

$$F(\alpha) \cong 1 + \alpha + \text{list}(\alpha) \times F(\alpha^2+\alpha^3) \times \text{list}(\alpha)$$

(finger trees)

list(α) may be optimized to $\alpha + \alpha^2 + \alpha^3 + \alpha^4$

Non-regular Recursion and Change of Bases / Chain Rules

$$F(\alpha) \cong 1 + \alpha + \text{list}(\alpha) \times F(\alpha^2+\alpha^3) \times \text{list}(\alpha)$$

(finger trees)

$$F'(\alpha) \cong 1 \cup (\mathbb{N} + F'(\alpha^2+\alpha^3) \times (2 \cup 3) + \mathbb{N})$$

list(α) may be optimized to $\alpha + \alpha^2 + \alpha^3 + \alpha^4$

Corollary of Conjecture

$$p : \forall\alpha.\ (F(\alpha)\to\alpha)\times(G(\alpha)\to K)\to H(\alpha)$$

Corollary of Conjecture

$$p : \forall \alpha.\ (F(\alpha) \to \alpha) \times (G(\alpha) \to K) \to H(\alpha)$$

$$\log_\alpha((F(\alpha)\to\alpha)\times(G(\alpha)\to K))$$
$$= \underbrace{\log_\alpha(F(\alpha)\to\alpha)}_{= F(\alpha)} + \underbrace{\log_\alpha(G(\alpha)\to K)}_{\approx 0}$$

$$\alpha^* = \mu\alpha.(F(\alpha) + 0) \cong \mu F$$

Corollary of Conjecture

$$p : \forall\alpha.\ (F(\alpha)\to\alpha)\times(G(\alpha)\to K)\to H(\alpha)$$

$$\log_\alpha((F(\alpha)\to\alpha)\times(G(\alpha)\to K))$$
$$= \underbrace{\log_\alpha(F(\alpha)\to\alpha)}_{=F(\alpha)} + \underbrace{\log_\alpha(G(\alpha)\to K)}_{\approx 0}$$

$$\alpha^* = \mu\alpha.(F(\alpha) + 0) \cong \mu F$$

How to plug in ρ^*, the catamorphism ("roll")?

Upgraded Logarithmic Conjecture

$$p : \forall\alpha.\ P(\alpha) \to H(\alpha)$$

POS

$$\alpha^* = \mu\alpha.P'(\alpha)$$ *if it exists*

$$\rho^+ : P^-(\alpha^*) \to P(\alpha^*)$$

to utilize

"roll" $\rho^* : P'(\alpha^*) \to \alpha^*$

$$\rho^+ : P^-(\alpha^*) \to P(\alpha^*)$$

to utilize

"roll" $\rho^* : P'(\alpha^*) \to \alpha^*$

ρ^* fixes arguments
at strictly positive locations

$P^-(\alpha^*)$: the residual

Killing the Strictly Positive

$$\alpha^{-} = 1 \qquad K^{-} = K$$
$$(A \to B)^{-} = A \to B^{-}$$

$$(A \times B)^{-} = A^{-} \times B- \qquad (A \cup B)^{-} = A^{-} \cup B^{-}$$
$$(A + B)^{-} = A^{-} + B^{-} \qquad (\mu\beta.A)^{-} = \mu\beta.A^{-}$$

$$\rho^{+} : P^{-}(\alpha^{*}) \to P(\alpha^{*})$$

residual full

Corollary of *Upgraded* Conjecture

$$p : \forall\alpha.\ (F(\alpha)\to\alpha)\times(G(\alpha)\to K)\to H(\alpha)$$

$$((F(\alpha)\to\alpha) \times (G(\alpha)\to K))^{-}$$
$$= \underbrace{(F(\alpha)\to 1)}_{\cong 1} \times (G(\alpha)\to K)$$

$$\rho^{+}(\_, g) = (\rho^{*}, g)$$

with appropriate rules (omitted)

ρ^{+} fills in the ρ^{*} for you

Problems: Suboptimal Types

$$\text{exists} : \forall\alpha.(\alpha\rightarrow 2)\rightarrow \text{list}(\alpha)\rightarrow 2$$

$\alpha^* = \mathbb{N}$, but $\alpha = 2$ with id seems better

similarly “all”, “toString”, “filter”, etc.

Problems: Suboptimal Types

$$\text{exists} : \forall\alpha.(\alpha\to 2)\to\text{list}(\alpha)\to 2$$

$\alpha^* = \mathbb{N}$, but $\alpha = 2$ with id seems better

similarly "all", "toString", "filter", etc.

$$\text{length} : \forall\alpha.\text{list}(\alpha)\to\mathbb{N}$$

$\alpha^* = \mathbb{N}$, but $\alpha = 1$ suffices

Problems: Suboptimal Types

$\text{exists} : \forall\alpha.(\alpha\to 2)\to \text{list}(\alpha)\to 2$

$\alpha^* = \mathbb{N}$, but $\alpha = 2$ with id seems better

similarly “all”, “toString”, “filter”, etc.

$\text{length} : \forall\alpha.\text{list}(\alpha)\to\mathbb{N}$

$\alpha^* = \mathbb{N}$, but $\alpha = 1$ suffices

Current: full distinctiveness

Next: *inspectability*

Future Work

Future Work

Conjectures → Theorems

Future Work

Conjectures → Theorems

Greatest fixed points, etc.

Future Work

Conjectures → Theorems

Greatest fixed points, etc.

Unknown *datatypes,* not just functions

Checker for parts of CMU 15-210 in 2015
General theory under development

Some Related Work

Testing Polymorphic Properties

Jean-Philippe Bernardy, Patrik Jansson, and Koen Claessen

Categories of Containers

Michael Abbott, Thorsten Altenkirch, and Neil Ghani

A semantics for shape

C. Barry Jay

Species and Functors and Types, Oh My!

Brent A. Yorgey

Unlike derivatives,
logarithms differentiate.

credit to Danny Gratzer