From equations to distinctions: Two interpretations of effectful computations

Niels Voorneveld

Tallinn University of Technology

August 31st, 2020
MSFP 2020

European Union
European Social Fund

Investing
in your future

Introduction

Two sides to program equivalence. For any two programs, either:

- the two programs 'simulate' eachother,
- or there is an observable difference between the two programs.

Program equivalence for programs with algebraic effects starts with specifying the behaviour of effects at base type in one of two ways:

- choosing axiomatic equations and inequations,
- or formulating distinction tests by EM-algebra.

	Equations	Distinctions
Basic relation	Equational axioms	EM-algebra
Higher-order	Applicative bisimilarity	Quantitative logic

Effect operations

Algebraic effect operations:

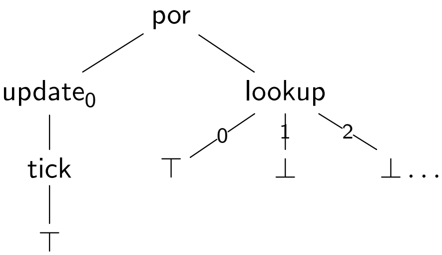
Probabilistic choice	$\mathsf{por} : 2$	
Global store	$\mathsf{lookup} : \mathbb{N}$	$\mathsf{update}_n : 1$
Cost counter	$\mathsf{tick} : 1$	

Property satisfaction tokens:

Failure (e.g. divergence)	$\bot : 0$
Success (e.g. final termination)	$\top : 0$

Effect trees

Possibly infinite effects trees (monad T):



Monad in the category of sets and preorders.
Syntactic order where:

$$\forall t \in TX.\ \bot \leq t \leq \top$$

Equations

Given a countable set of variables $\mathbb{V}$, a tree $e \in T\mathbb{V}$ is an *algebraic expression*.

Axiomatic effect-specific (in)equations A $\subseteq (T\mathbb{V})^2$

Probability:

$$\begin{aligned} \mathrm{por}(x,x) &= x \\ \mathrm{por}(x,y) &= \mathrm{por}(y,x) \\ \mathrm{por}(\mathrm{por}(x,y),\mathrm{por}(z,w)) &= \mathrm{por}(\mathrm{por}(x,z),\mathrm{por}(y,w)) \\ \mathrm{por}(x,\mathrm{por}(x,\mathrm{por}(x,..))) &= x \end{aligned}$$

Global store:

$$\begin{aligned} \mathrm{up}_i(\mathrm{up}_j(x)) &= \mathrm{up}_j(x) \\ \mathrm{up}_i(\mathrm{lo}(x_0,x_1,\dots)) &= \mathrm{up}_i(x_i) \\ \mathrm{lo}(\mathrm{up}_0(x_0),\mathrm{up}_1(x_1),\dots) &= \mathrm{lo}(x_0,x_1,\dots) \\ \mathrm{lo}(x,x,\dots) &= x \end{aligned}$$

Algebraic relation

Induced *algebraic relation* $\mathcal{R}_{\mathsf{A}} \subseteq (T\mathbb{V})^2$:

- Reflexive and transitive
- Compositional (and substitutional)
- Preserves syntactic order
- Admissable

Simple derivation in global store:

$$\mathsf{update}_i(\top) \quad \geq \quad \mathsf{update}_i(\mathsf{update}_j(\top)) \quad = \quad \mathsf{update}_j(\top)$$

$$\begin{aligned} \top &= \mathsf{lookup}(\top, \top, \top, \dots) \\ &= \mathsf{lookup}(\mathsf{update}_0(\top), \mathsf{update}_1(\top), \mathsf{update}_2(\top), \dots) \\ &= \mathsf{lookup}(\mathsf{update}_i(\top), \mathsf{update}_i(\top), \mathsf{update}_i(\top), \dots) \\ &= \mathsf{update}_i(\top) \end{aligned}$$

Algebras

Definition

An algebra $\alpha : T\mathbb{O} \to \mathbb{O}$ of a monad $T = (T, \eta, \mu)$ is an *EM-algebra* if:

$$\begin{array}{ccc} \mathbb{O} & \xrightarrow{\eta_{\mathbb{O}}} & T\mathbb{O} \\ & \diagdown\!\!\diagdown & \downarrow{\scriptstyle \alpha} \\ & & \mathbb{O} \end{array} \qquad \begin{array}{ccc} TT\mathbb{O} & \xrightarrow{T\alpha} & T\mathbb{O} \\ {\scriptstyle \mu_{\mathbb{O}}}\downarrow & & \downarrow{\scriptstyle \alpha} \\ T\mathbb{O} & \xrightarrow[\alpha]{} & \mathbb{O} \end{array}$$

We work in the category of preorders, with truth space $\mathbb{O}$.
The algebra lifts each predicate $P : X \to \mathbb{O}$ to $\alpha \circ TP : TX \to \mathbb{O}$.

Induces an algebraic relation $\mathcal{R}_\alpha \subseteq (T\mathbb{V})^2$ as follows:

$$e_1 \ \mathcal{R}_\alpha \ e_2 \iff \forall P : \mathbb{V} \to \mathbb{O}.\ (\alpha \circ TP)(e_1) \leq (\alpha \circ TP)(e_2)$$

Equations $\Longrightarrow$ algebras

Suppose we have an algebraic relation $\mathcal{I} \subseteq (T\mathbb{V})^2$.

Let $\mathcal{B} := \mathcal{I} \cap (T\emptyset)^2$ be the sub-relation on fundamental (continuation free) computations.

Consider the quotient $[\mathcal{B}] := T\emptyset/\mathcal{B}$ with injection $\iota : T\emptyset \to [\mathcal{B}]$.

Extracting an algebra:

$$\begin{array}{ccc} T[\mathcal{B}] & \overset{\alpha}{\dashrightarrow} & [\mathcal{B}] \\ {\scriptstyle T\iota}\uparrow & & \uparrow{\scriptstyle \iota} \\ TT\emptyset & \underset{\mu^T_\emptyset}{\longrightarrow} & T\emptyset \end{array}$$

Lemma

If $\mathcal{I}$ is reflexive, transitive and compositional, then $\alpha_{\mathcal{I}}$ is a well-defined EM-algebra.

Algebra for probability

Consider the axioms A for probability.

For probability, $T\emptyset$ are binary trees whose leaves are either $\bot$ or $\top$.
$\forall e_1, e_2 \in T\emptyset, \quad e_1 \mathcal{R}_{\mathrm{A}} e_2 \quad \Leftrightarrow \quad \mathrm{P}(e_1 \mapsto \top) \leq \mathrm{P}(e_2 \mapsto \top),$
so $[\mathcal{B}] = [0, 1]$ the real number interval.

Derived algebra is the expectations function $\mathrm{Exp} : T[0,1] \to [0,1]$,
where $\quad \mathrm{Exp}(\mathrm{por}(x,y)) = (\mathrm{Exp}(x) + \mathrm{Exp}(y))/2.$

Implements the fundamental observational question:
What is the probability that the program ...?

Algebra for global store

Consider the axioms A for global store.

$\forall e \in T\emptyset, \exists! S \subseteq \mathbb{N}$: $e\ \mathcal{R}_{\mathsf{A}}$ lookup$(0 \in S, 1 \in S, 2 \in S, \dots)$.
so $[\mathcal{B}] := \mathcal{P}(\mathbb{N}) \simeq \mathbb{N} \to \mathbb{B}$, the order of state predicates.

The constructed algebra is the weakest precondition function $\mathsf{Wp} : T(\mathcal{P}(\mathbb{N})) \to \mathcal{P}(\mathbb{N})$ where,

$$\mathsf{Wp}(up_i(x)) = \{n \in \mathbb{N} \mid i \in \mathsf{Wp}(x)\}$$
$$\mathsf{Wp}(lo(x_0, x_1, \dots)) = \{n \in \mathbb{N} \mid n \in \mathsf{Wp}(x_n)\}$$

Implements the fundamental observational question:
For which starting states does the program ...?

Coincidence between algebra and equations

Lemma

$\mathcal{I}$ reflexive, transitive and compositional $\mathcal{I} \subseteq \mathcal{R}_{\alpha_\mathcal{I}}$.

Definition

$\mathcal{I}$ is *base-valued* if for any $e_1, e_2 \in T\mathbb{V}$,

$$e_1 \ \mathcal{I} \ e_2 \iff \forall P : \mathbb{V} \to T\emptyset.\ P^*(e_1) \ \mathcal{I} \ P^*(e_2)$$

Proposition

If $\mathcal{I}$ *is reflexive, transitive, compositional and base-valued, then* $\mathcal{I} = \mathcal{R}_{\alpha_\mathcal{I}}$.

Program equivalence: Quantitative logic

Going higher-order: Program equivalence/approximation.

We use $\mathbb{O}$ as the truth space for quantitative predicates.
If $\mathbb{O}$ is a complete lattice, Quantitative Distinction Logic:

$$\frac{n \in \mathbb{N}}{\{n\} \in \mathit{formulas}(\mathbb{N})} \qquad \frac{\underline{\phi} \in \mathit{formulas}(\underline{C})}{U(\underline{\phi}) \in \mathit{formulas}(U(\underline{C}))}$$

$$\frac{V{:}A \quad \underline{\phi} \in \mathit{formulas}(\underline{C})}{V{\mapsto}\underline{\phi} \in \mathit{formulas}(\underline{C})} \qquad \frac{\phi \in \mathit{formulas}(A)}{\alpha(\phi) \in \mathit{formulas}(F(A))}$$

$$\frac{\Phi \subseteq \mathit{formulas}(\overline{E})}{\bigvee \Phi \in \mathit{formulas}(\overline{E})} \qquad \frac{\Phi \subseteq \mathit{formulas}(\overline{E})}{\bigwedge \Phi \in \mathit{formulas}(\overline{E})}$$

$$\frac{a \in \mathbb{O}}{\mathrm{constant}(a) \in \mathit{formulas}(\overline{E})} \qquad \frac{a \in \mathbb{O} \quad \overline{\phi} \in \mathit{formulas}(\overline{E})}{(\overline{\phi}{\geq}a) \in \mathit{formulas}(\overline{E})}$$

This is an example of such a logic for a CBPV language.

Program equivalence: Applicative simulation

On the equating side: Applicative bisimilarity/similarity.

If $\mathbb{O}$ is a complete lattice, we can define a relator
$\Gamma^{\alpha}_{X,Y} : Relation(X,Y) \to Relation(TX,TY)$

$\Gamma^{\alpha}_{X,Y}(\mathcal{R})$ is the largest relation s.t.
for all $P : X \to \mathbb{O}$ and $Q : Y \to \mathbb{O}$:

$$\begin{array}{ccc} X & \overset{\mathcal{R}}{\longrightarrow\!\!\!\!\!\!|\,\,\,} & Y \\ {\scriptstyle P}\downarrow & & \downarrow{\scriptstyle Q} \\ \mathbb{O} & \underset{\leq}{\longrightarrow\!\!\!\!\!\!|\,\,\,} & \mathbb{O} \end{array} \quad \Longrightarrow \quad \begin{array}{ccc} TX & \overset{\Gamma^{\alpha}_{X,Y}(\mathcal{R})}{\longrightarrow\!\!\!\!\!\!|\,\,\,} & TY \\ {\scriptstyle \alpha(P)}\downarrow & & \downarrow{\scriptstyle \alpha(Q)} \\ \mathbb{O} & \underset{\leq}{\longrightarrow\!\!\!\!\!\!|\,\,\,} & \mathbb{O} \end{array}$$

Two programs are related if one “simulates” the other.
Applicative simulations generate program approximation.

Algebra for cost

An operation for associating cost: $\mathsf{tick} : 1$.

$$\mathsf{tick}(x) \;\leq\; x$$

Simple derivation:

$$\bot \;\leq\; \mathsf{tick}(\bot) \;\leq\; \bot \quad\Longrightarrow\quad \bot = \mathsf{tick}(\bot)$$

Carrier $[\mathcal{B}] = \mathbb{N} + \{\infty\}$.
Algebra is the Tally operation: $\mathsf{Tally}(\mathsf{tick}(x)) = \mathsf{Tally}(x) + 1$.

Implements the fundamental observational question:
For which price does the program ...?

Algebra for effect combinations

Probability + Global store, operations distribute:

$$\mathsf{update}_i(\mathsf{por}(x,y)) \quad = \quad \mathsf{por}(\mathsf{update}_i(x),\mathsf{update}_i(y))$$

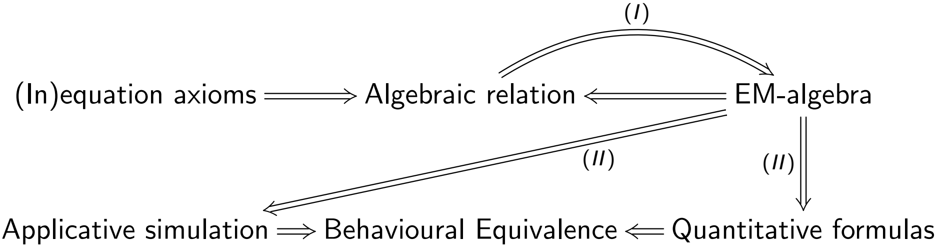
$$\mathsf{por}(\mathsf{lookup}(x_0,x_1,\dots),y) \quad = \quad \mathsf{lookup}(\mathsf{por}(x_0,y),\mathsf{por}(x_1,y),\dots)$$

For all $r \in [0,1]$ let $\langle r\rangle \in T\emptyset$ s.t. $\mathsf{P}(\langle r\rangle \mapsto \top) = r$.
$\forall e \in T\emptyset, \exists! p : \mathbb{N} \to [0,1]$, $e\ \mathcal{R}_{\mathsf{A}}\ \mathsf{lookup}(\langle p(0)\rangle, \langle p(1)\rangle, \langle p(2)\rangle, \dots)$

Carrier: $\mathbb{N} \to [0,1]$.
Algebra: Weakest probabilistic precondition.

Implements the fundamental observational question:
For each starting state, what is the probability that the program...?

Conclusion



(I) Current work: '1-to-1' construction for base-valued relations.

(II) Previous work: Formulation of a congruent program equivalence if $\mathbb{O}$ is a complete lattice and algebra is ω-continuous.