Affine Monads and Side-Effect-Freeness

CMCS 2016, Eindhoven

Bart Jacobs
bart@cs.ru.nl
2 April 2016

iCIS | Digital Security
Radboud University

Outline

Where we are, sofar

Context: effectus theory and side-effects

Affine monads

Predicates and instruments

Main results

Conclusions

Quantum

Quantum

- Quantum computation and logic is a fascinating area
- "Hot topic", because of all the buzz about quantum computers
- Potentially large impact, esp. in security
  - existing (public key) algorithms are vulnerable
  - new research area: "post-quantum crypto"
- New challenges for existing concepts in (theoretical) CS
  - three overlapping areas: physics, math, CS
  - John Baez: category theory is "Rosetta Stone"
- Strong coalgebraic flavour
  - "states" play an important role
  - quantum observations can have a side-effect (state-change)

Logic, side-effects, and commutativity

Logic, side-effects, and commutativity

Consider the logical equivelence $\equiv$ of:

$$\text{it's raining} \wedge \text{Ichiro is sleeping} \;\equiv\; \text{Ichiro is sleeping} \wedge \text{it's raining}$$

Logic, side-effects, and commutativity

Consider the logical equivelence $\equiv$ of:

$$\text{it's raining} \wedge \text{Ichiro is sleeping} \;\equiv\; \text{Ichiro is sleeping} \wedge \text{it's raining}$$

Conjunction $\wedge$ is obviously commutative

Logic, side-effects, and commutativity

Consider the logical equivelence $\equiv$ of:

it's raining $\wedge$ Ichiro is sleeping $\equiv$ Ichiro is sleeping $\wedge$ it's raining

Conjunction $\wedge$ is obviously commutative

Compare this to:

there are 5 eggs in the basket $\wedge$ Ichiro is making an omelette

$\overset{??}{\equiv}$ Ichiro is making an omelette $\wedge$ there are 5 eggs in the basket

Logic, side-effects, and commutativity

Consider the logical equivelence $\equiv$ of:

it's raining $\wedge$ Ichiro is sleeping $\equiv$ Ichiro is sleeping $\wedge$ it's raining

Conjunction $\wedge$ is obviously commutative

Compare this to:

there are 5 eggs in the basket $\wedge$ Ichiro is making an omelette

$\overset{??}{\equiv}$ Ichiro is making an omelette $\wedge$ there are 5 eggs in the basket

- If predicates can have side-effects, commutativity is no longer obvious. Conjunction should be used as 'and-then'

Logic, side-effects, and commutativity

Consider the logical equivelence $\equiv$ of:

it's raining $\wedge$ Ichiro is sleeping $\equiv$ Ichiro is sleeping $\wedge$ it's raining

Conjunction $\wedge$ is obviously commutative

Compare this to:

there are 5 eggs in the basket $\wedge$ Ichiro is making an omelette

$\stackrel{??}{\equiv}$ Ichiro is making an omelette $\wedge$ there are 5 eggs in the basket

- If predicates can have side-effects, commutativity is no longer obvious. Conjunction should be used as 'and-then'
- This plays an important role in the quantum world — and also in imperative programming where & (and &&) are not commutative

Effectus theory

Effectus theory

- Own (group's) work has led to a new categorical notion: effectus
  - it's a certain kind of category, with $0, +, 1$, some pullbacks, and some jointly monic maps
  - its predicates form effect modules, its states are convex sets, and together they form a "state-and-effect" triangle

Effectus theory

- Own (group's) work has led to a new categorical notion: effectus
  - it's a certain kind of category, with $0, +, 1$, some pullbacks, and some jointly monic maps
  - its predicates form effect modules, its states are convex sets, and together they form a "state-and-effect" triangle
- An effectus is an abstract model for quantum computation and logic
  - probabilistic computation forms a special "commutative" subclass
  - Boolean computation is a further "idempotent" subclass

Effectus theory

- Own (group's) work has led to a new categorical notion: effectus
  - it's a certain kind of category, with $0, +, 1$, some pullbacks, and some jointly monic maps
  - its predicates form effect modules, its states are convex sets, and together they form a "state-and-effect" triangle
- An effectus is an abstract model for quantum computation and logic
  - probabilistic computation forms a special "commutative" subclass
  - Boolean computation is a further "idempotent" subclass
- Side-effects are part of the formalism, via instruments

Effectus theory

- Own (group's) work has led to a new categorical notion: effectus
  - it's a certain kind of category, with $0, +, 1$, some pullbacks, and some jointly monic maps
  - its predicates form effect modules, its states are convex sets, and together they form a "state-and-effect" triangle
- An effectus is an abstract model for quantum computation and logic
  - probabilistic computation forms a special "commutative" subclass
  - Boolean computation is a further "idempotent" subclass
- Side-effects are part of the formalism, via instruments

  For each predicate p on X, there is an instrument map:

$$X \xrightarrow{\mathrm{instr}_p} X + X$$

Effectus theory

- Own (group's) work has led to a new categorical notion: effectus
  - it's a certain kind of category, with $0, +, 1$, some pullbacks, and some jointly monic maps
  - its predicates form effect modules, its states are convex sets, and together they form a "state-and-effect" triangle
- An effectus is an abstract model for quantum computation and logic
  - probabilistic computation forms a special "commutative" subclass
  - Boolean computation is a further "idempotent" subclass
- Side-effects are part of the formalism, via instruments

  For each predicate p on X, there is an instrument map:

$$X \xrightarrow{\mathrm{instr}_p} X + X$$

  It is called side-effect-free if $\nabla \circ \mathrm{instr}_p = \mathrm{id}$, where $\nabla = [\mathrm{id}, \mathrm{id}]$.

iCIS | Digital Security
Radboud University

Effectus theory

- Own (group's) work has led to a new categorical notion: effectus
  - it's a certain kind of category, with $0, +, 1$, some pullbacks, and some jointly monic maps
  - its predicates form effect modules, its states are convex sets, and together they form a "state-and-effect" triangle
- An effectus is an abstract model for quantum computation and logic
  - probabilistic computation forms a special "commutative" subclass
  - Boolean computation is a further "idempotent" subclass
- Side-effects are part of the formalism, via instruments

  For each predicate p on X, there is an instrument map:

  $$X \xrightarrow{\mathrm{instr}_p} X + X$$

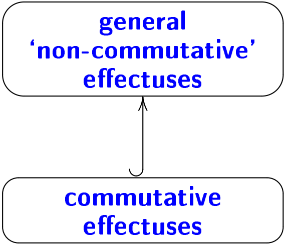
  It is called side-effect-free if $\nabla \circ \mathrm{instr}_p = \mathrm{id}$, where $\nabla = [\mathrm{id}, \mathrm{id}]$.
- **We have**: in the probabilistic and Boolean case, instruments are side-effect-free, but not in the quantum case!

Overview: subclasses of effectuses (ArXiv, 1512.05813)

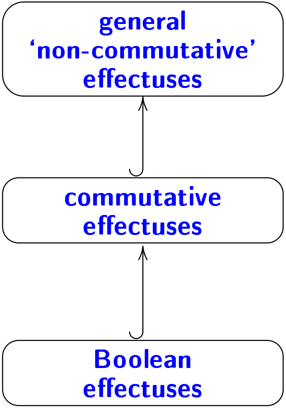
Overview: subclasses of effectuses (ArXiv, 1512.05813)

general
'non-commutative'
effectuses

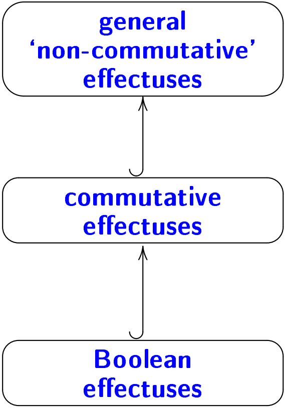
Overview: subclasses of effectuses (ArXiv, 1512.05813)



Overview: subclasses of effectuses (ArXiv, 1512.05813)

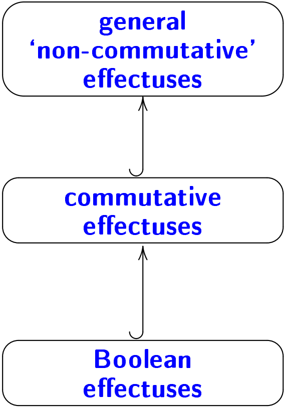


Overview: subclasses of effectuses (ArXiv, 1512.05813)



von Neumann algebras $\mathbf{vNA}^{\mathrm{op}}$

Overview: subclasses of effectuses (ArXiv, 1512.05813)

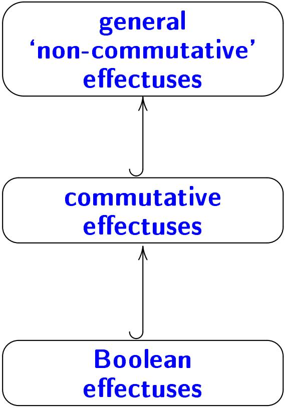


von Neumann algebras $\mathbf{vNA}^{\mathrm{op}}$

commutative von Neumann algebras,

$\mathcal{K}\ell(\mathcal{D}), \mathcal{K}\ell(\mathcal{G}), \ldots$

Overview: subclasses of effectuses (ArXiv, 1512.05813)



von Neumann algebras $\mathbf{vNA}^{\mathrm{op}}$

commutative von Neumann algebras,

$\mathcal{K}\ell(\mathcal{D}), \mathcal{K}\ell(\mathcal{G}), \ldots$

Sets,

extensive categories

Characterising subclasses

Characterising subclasses

Theorem (See Effectus Intro paper on ArXiv)

The Boolean effectuses are precisely the extensive categories (with 1*).*

iCIS | Digital Security
Radboud University

Characterising subclasses

Theorem (See Effectus Intro paper on ArXiv)

The Boolean effectuses are precisely the extensive categories (with 1*).*

Wild conjecture

Commutative effectuses are Kleisli categories of a commutative monad

Characterising subclasses

Theorem (See Effectus Intro paper on ArXiv)

The Boolean effectuses are precisely the extensive categories (with 1).

Wild conjecture

Commutative effectuses are Kleisli categories of a commutative monad

Examples: $\mathcal{K}\ell(\mathcal{D}) \quad \mathcal{K}\ell(\mathcal{G}) \quad \mathcal{K}\ell(\mathcal{E}) \quad \mathcal{K}\ell(\mathcal{R}) \simeq \mathbf{CvNA}^{\mathrm{op}} \quad \ldots$

iCIS | Digital Security
Radboud University

Main question underlying the CMCS paper

iCIS | Digital Security
Radboud University

Main question underlying the CMCS paper

How are effectus properties and monad properties connected?

Main question underlying the CMCS paper

How are effectus properties and monad properties connected?

- Is there a relation between commutativity in effectuses and commutativity of monads?

Main question underlying the CMCS paper

How are effectus properties and monad properties connected?

- Is there a relation between commutativity in effectuses and commutativity of monads?
- Is side-effect-freeness related to some property of a monad

Main question underlying the CMCS paper

How are effectus properties and monad properties connected?

- Is there a relation between commutativity in effectuses and commutativity of monads?
- Is side-effect-freeness related to some property of a monad
  - being "affine" is a candidate — that is, $T(1) \cong 1$

Main question underlying the CMCS paper

How are effectus properties and monad properties connected?

- Is there a relation between commutativity in effectuses and commutativity of monads?
- Is side-effect-freeness related to some property of a monad
  - being "affine" is a candidate — that is, $T(1) \cong 1$

These questions have "good" answers

- they are first steps towards the *wild conjecture*

Where we are, sofar

Setting

Setting

- We work in a distributive category **C**
  - with finite products $(1, \times)$ and coproducts $(0, +)$,
  - where $\times$ distributes over $+$

Setting

- We work in a distributive category $\mathbf{C}$
  - with finite products $(1, \times)$ and coproducts $(0, +)$,
  - where $\times$ distributes over $+$
- We assume a monad $T\colon \mathbf{C} \to \mathbf{C}$

Setting

- We work in a distributive category $\mathbf{C}$
  - with finite products $(1, \times)$ and coproducts $(0, +)$,
  - where $\times$ distributes over $+$
- We assume a monad $T\colon \mathbf{C} \to \mathbf{C}$
- The monad is strong if there is a strength map $\mathrm{st}_1\colon T(X) \times Y \to T(X \times Y)$ suitably commuting with other structure
  - by swapping we get $\mathrm{st}_2\colon X \times T(Y) \to T(X \times Y)$

Setting

- We work in a distributive category $\mathbf{C}$
  - with finite products $(1, \times)$ and coproducts $(0, +)$,
  - where $\times$ distributes over $+$
- We assume a monad $T\colon \mathbf{C} \to \mathbf{C}$
- The monad is strong if there is a strength map $\mathrm{st}_1\colon T(X) \times Y \to T(X \times Y)$ suitably commuting with other structure
  - by swapping we get $\mathrm{st}_2\colon X \times T(Y) \to T(X \times Y)$
- The monad is commutative if the following diagram commutes:

$$\begin{array}{ccccccc}
 & \overset{\mathrm{st}_1}{\nearrow} & T(X \times T(Y)) & \xrightarrow{T(\mathrm{st}_2)} & T^2(X \times Y) & \overset{\mu}{\searrow} & \\
T(X) \times T(Y) & & & & & & T(X \times Y) \\
 & \underset{\mathrm{st}_2}{\searrow} & T(T(X) \times Y) & \xrightarrow[T(\mathrm{st}_1)]{} & T^2(X \times Y) & \underset{\mu}{\nearrow} &
\end{array}$$

Affineness

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

- The non-empty powerset monad $\mathcal{P}_+$ on **Sets**

iCIS | Digital Security
Radboud University

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

- The non-empty powerset monad $\mathcal{P}_+$ on **Sets**
- The distribution monad $\mathcal{D}$ on **Sets**

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

- The non-empty powerset monad $\mathcal{P}_+$ on **Sets**
- The distribution monad $\mathcal{D}$ on **Sets**
- The Giry monad $\mathcal{G}$ on **Meas**

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

- The non-empty powerset monad $\mathcal{P}_+$ on **Sets**
- The distribution monad $\mathcal{D}$ on **Sets**
- The Giry monad $\mathcal{G}$ on **Meas**
- The expectation monad $\mathcal{E} = \mathbf{EMod}\big([0,1]^{(-)}, [0,1]\big)$ on **Sets**

iCIS | Digital Security
Radboud University

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

- The non-empty powerset monad $\mathcal{P}_+$ on **Sets**
- The distribution monad $\mathcal{D}$ on **Sets**
- The Giry monad $\mathcal{G}$ on **Meas**
- The expectation monad $\mathcal{E} = \mathbf{EMod}\big([0,1]^{(-)}, [0,1]\big)$ on **Sets**
- The Radon monad $\mathcal{R} = \mathit{Stat}(C(-))$ on **CH**

Affineness

Definition

The monad T is called affine if $T(1) \cong 1$

Examples

- The non-empty powerset monad $\mathcal{P}_+$ on **Sets**
- The distribution monad $\mathcal{D}$ on **Sets**
- The Giry monad $\mathcal{G}$ on **Meas**
- The expectation monad $\mathcal{E} = \mathbf{EMod}\big([0,1]^{(-)}, [0,1]\big)$ on **Sets**
- The Radon monad $\mathcal{R} = \mathit{Stat}(C(-))$ on **CH**

Note: if T is affine, then 1 is final in $\mathcal{K}\ell(T)$.

Affine submonad

Affine submonad

Assuming enough pullbacks, the affine submonad $T_a \rightarrowtail T$ is defined via:

$$\begin{array}{ccc} T_a(X) & \longrightarrow & 1 \\ \big\downarrow & \lrcorner & \big\downarrow \eta \\ T(X) & \xrightarrow[T(!)]{} & T(1) \end{array}$$

Affine submonad

Assuming enough pullbacks, the affine submonad $T_a \rightarrowtail T$ is defined via:

$$\begin{array}{ccc} T_a(X) & \longrightarrow & 1 \\ \big\downarrow & \lrcorner & \big\downarrow{\scriptstyle \eta} \\ T(X) & \xrightarrow[T(!)]{} & T(1) \end{array}$$

Lemma (Lindner 1979)

Affine submonad

Assuming enough pullbacks, the affine submonad $T_a \rightarrowtail T$ is defined via:

$$\begin{array}{ccc} T_a(X) & \longrightarrow & 1 \\ \big\downarrow & \lrcorner & \big\downarrow{\scriptstyle \eta} \\ T(X) & \xrightarrow[T(!)]{} & T(1) \end{array}$$

Lemma (Lindner 1979)

- *This T_a is an affine monad, and $T_a \rightarrowtail T$ is a monad map*

Affine submonad

Assuming enough pullbacks, the affine submonad $T_a \rightarrowtail T$ is defined via:

$$\begin{array}{ccc} T_a(X) & \longrightarrow & 1 \\ \big\downarrow & \lrcorner & \big\downarrow{\scriptstyle \eta} \\ T(X) & \xrightarrow[T(!)]{} & T(1) \end{array}$$

Lemma (Lindner 1979)

- *This T_a is an affine monad, and $T_a \rightarrowtail T$ is a monad map*
  - *in fact, T_a is the greatest affine submonad*

Affine submonad

Assuming enough pullbacks, the affine submonad $T_a \rightarrowtail T$ is defined via:

$$\begin{array}{ccc} T_a(X) & \longrightarrow & 1 \\ \downarrow & \lrcorner & \downarrow{\scriptstyle \eta} \\ T(X) & \xrightarrow[T(!)]{} & T(1) \end{array}$$

Lemma (Lindner 1979)

- *This T_a is an affine monad, and $T_a \rightarrowtail T$ is a monad map*
  - *in fact, T_a is the greatest affine submonad*
- *if T is strong / commutative then so is T_a*

iCIS | Digital Security
Radboud University

Affine submonad examples

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\mathcal{P}_a(X)$$

iCIS | Digital Security
Radboud University

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\mathcal{P}_a(X) = \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\}$$

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned}\mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\}\end{aligned}$$

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned}\mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid \{* \mid x \in U\} = \{*\}\}\end{aligned}$$

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned}\mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid \{* \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid U \neq \emptyset\}\end{aligned}$$

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned}\mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid \{* \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid U \neq \emptyset\}\end{aligned}$$

- The affine submonad of multiset monad $\mathcal{M}_{\mathbb{R}_{\geq 0}}$ is distribution $\mathcal{D}$

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned} \mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid \{* \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid U \neq \emptyset\} \end{aligned}$$

- The affine submonad of multiset monad $\mathcal{M}_{\mathbb{R}_{\geq 0}}$ is distribution $\mathcal{D}$
  We now restrict to formal sums $\varphi = \sum_i r_i | x_i \rangle$ with:

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned}\mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid \{* \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid U \neq \emptyset\}\end{aligned}$$

- The affine submonad of multiset monad $\mathcal{M}_{\mathbb{R}_{\geq 0}}$ is distribution $\mathcal{D}$

  We now restrict to formal sums $\varphi = \sum_i r_i| x_i \rangle$ with:

$$1 = \mathcal{M}(!)(\varphi)$$

Affine submonad examples

- The affine submonad of powerset is non-empty powerset

$$\begin{aligned}\mathcal{P}_a(X) &= \{U \subseteq X \mid \mathcal{P}(!)(U) = \{*\}\} \\ &= \{U \subseteq X \mid \{!(x) \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid \{* \mid x \in U\} = \{*\}\} \\ &= \{U \subseteq X \mid U \neq \emptyset\}\end{aligned}$$

- The affine submonad of multiset monad $\mathcal{M}_{\mathbb{R}_{\geq 0}}$ is distribution $\mathcal{D}$

  We now restrict to formal sums $\varphi = \sum_i r_i |\, x_i \,\rangle$ with:

$$1 = \mathcal{M}(!)(\varphi) = \sum_i r_i$$

iCIS | Digital Security
Radboud University

Causal maps

Causal maps

Write:

$$\bar{\top}_X \stackrel{\text{def}}{=} \left(X \xrightarrow{!_X} 1 \xrightarrow{\eta_1} T(1)\right)$$

Causal maps

Write:

$$\bar{\top}_X \stackrel{\text{def}}{=} \Big(X \xrightarrow{\ !_X\ } 1 \xrightarrow{\ \eta_1\ } T(1) \Big)$$

Definition

A map $f : X \to T(Y)$ is called causal if $\bar{\top}_Y \bullet f = \bar{\top}_X$, where $\bullet$ is Kleisli composition.

Causal maps

Write:

$$\bar{\top}_X \stackrel{\text{def}}{=} \Big(X \xrightarrow{\;!_X\;} 1 \xrightarrow{\;\eta_1\;} T(1) \Big)$$

Definition

A map $f : X \to T(Y)$ is called causal if $\bar{\top}_Y \bullet f = \bar{\top}_X$, where $\bullet$ is Kleisli composition.

Lemma

A map $X \to T(Y)$ is causal iff it factors as $X \to T_a(Y)$ via the affine submonad T_a

Causal maps

Write:

$$\bar{\top}_X \stackrel{\text{def}}{=} \Big(X \xrightarrow{!_X} 1 \xrightarrow{\eta_1} T(1) \Big)$$

Definition

A map $f : X \to T(Y)$ is called causal if $\bar{\top}_Y \bullet f = \bar{\top}_X$, where $\bullet$ is Kleisli composition.

Lemma

A map $X \to T(Y)$ is causal iff it factors as $X \to T_a(Y)$ via the affine submonad T_a

Example: maps $X \to \mathcal{D}(Y)$ are causal maps $X \to \mathcal{M}_{\mathbb{R}_{\geq 0}}(Y)$.

Where we are, sofar

Predicates

Predicates

- We shall work in the Kleisli category $\mathcal{K}\ell(T)$

iCIS | Digital Security
Radboud University

Predicates

- We shall work in the Kleisli category $\mathcal{K}\ell(T)$
- A predicate on X is a Kleisli map $X \to 2 = 1 + 1$
  - that is, a map $X \to T(1+1)$ in $\mathbf{C}$

Predicates

- We shall work in the Kleisli category $\mathcal{K}\ell(T)$
- A predicate on X is a Kleisli map $X \to 2 = 1 + 1$
  - that is, a map $X \to T(1+1)$ in $\mathbf{C}$
- There are truth and false predicates:

$$\mathbf{1} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_1} 2 \xrightarrow{\eta} T(2)\Big) \qquad \mathbf{0} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_2} 2 \xrightarrow{\eta} T(2)\Big)$$

iCIS | Digital Security
Radboud University

Predicates

- We shall work in the Kleisli category $\mathcal{K}\ell(T)$
- A predicate on X is a Kleisli map $X \to 2 = 1 + 1$
  - that is, a map $X \to T(1+1)$ in $\mathbf{C}$
- There are truth and false predicates:

$$\mathbf{1} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_1} 2 \xrightarrow{\eta} T(2)\Big) \qquad \mathbf{0} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_2} 2 \xrightarrow{\eta} T(2)\Big)$$

- There is also negation / orthosupplement

$$p^{\perp} = \Big(X \xrightarrow{p} T(1+1) \xrightarrow[\cong]{T([\kappa_2, \kappa_1])} T(1+1)\Big)$$

Predicates

- We shall work in the Kleisli category $\mathcal{K}\ell(T)$
- A predicate on X is a Kleisli map $X \to 2 = 1+1$
  - that is, a map $X \to T(1+1)$ in $\mathbf{C}$
- There are truth and false predicates:

$$\mathbf{1} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_1} 2 \xrightarrow{\eta} T(2)\Big) \qquad \mathbf{0} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_2} 2 \xrightarrow{\eta} T(2)\Big)$$

- There is also negation / orthosupplement

$$p^{\perp} = \Big(X \xrightarrow{p} T(1+1) \xrightarrow[\cong]{T([\kappa_2,\kappa_1])} T(1+1)\Big)$$

Note: $p^{\perp\perp} = p$ and $\mathbf{1}^{\perp} = \mathbf{0}$ and $\mathbf{0}^{\perp} = \mathbf{1}$

Predicates

- We shall work in the Kleisli category $\mathcal{K}\ell(T)$
- A predicate on X is a Kleisli map $X \to 2 = 1 + 1$
  - that is, a map $X \to T(1+1)$ in $\mathbf{C}$
- There are truth and false predicates:

$$\mathbf{1} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_1} 2 \xrightarrow{\eta} T(2)\Big) \qquad \mathbf{0} = \Big(X \longrightarrow 1 \xrightarrow{\kappa_2} 2 \xrightarrow{\eta} T(2)\Big)$$

- There is also negation / orthosupplement

$$p^{\perp} = \Big(X \xrightarrow{p} T(1+1) \xrightarrow[\cong]{T([\kappa_2,\kappa_1])} T(1+1)\Big)$$

Note: $p^{\perp\perp} = p$ and $\mathbf{1}^{\perp} = \mathbf{0}$ and $\mathbf{0}^{\perp} = \mathbf{1}$

- In many (probabilistic) examples, predicates are maps $X \to [0,1]$

Instruments

iCIS | Digital Security
Radboud University

Instruments

For a predicate $p\colon X \to 1+1$ we define an instrument $\mathrm{instr}_p\colon X \to X + X$ in $\mathcal{K}\ell(T)$ as:

$$\mathrm{instr}_p = \Big(X \xrightarrow{\langle p, \mathrm{id}\rangle} T(2) \times X \xrightarrow{\mathrm{st}_1} T(2 \times X) \xrightarrow{\cong} T(X + X)\Big)$$

Instruments

For a predicate $p\colon X \to 1+1$ we define an instrument $\mathrm{instr}_p\colon X \to X+X$ in $\mathcal{K}\ell(T)$ as:

$$\mathrm{instr}_p = \Big(X \xrightarrow{\langle p, \mathrm{id}\rangle} T(2) \times X \xrightarrow{\mathrm{st}_1} T(2 \times X) \xrightarrow{\cong} T(X+X)\Big)$$

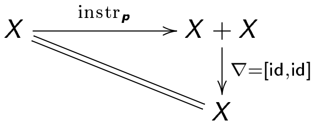
- We have $(! + !) \bullet \mathrm{instr}_p = p$

Instruments

For a predicate $p \colon X \to 1+1$ we define an instrument $\mathrm{instr}_p \colon X \to X + X$ in $\mathcal{K}\ell(T)$ as:

$$\mathrm{instr}_p = \Big(X \xrightarrow{\langle p, \mathrm{id} \rangle} T(2) \times X \xrightarrow{\mathrm{st}_1} T(2 \times X) \xrightarrow{\cong} T(X + X) \Big)$$

- We have $(! + !) \bullet \mathrm{instr}_p = p$
- The instrument is called side-effect-free if:

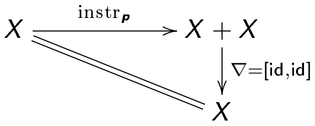


Instruments

For a predicate $p \colon X \to 1+1$ we define an instrument $\mathrm{instr}_p \colon X \to X + X$ in $\mathcal{K}\ell(T)$ as:

$$\mathrm{instr}_p = \Big(X \xrightarrow{\langle p, \mathrm{id} \rangle} T(2) \times X \xrightarrow{\mathrm{st}_1} T(2 \times X) \xrightarrow{\cong} T(X + X) \Big)$$

- We have $(! + !) \bullet \mathrm{instr}_p = p$
- The instrument is called side-effect-free if:



Lemma

If T is affine, then each instrument is side-effect-free

Instrument example: powerset

Instrument example: powerset

- Take a predicate $p\colon X \to \mathcal{P}(2) \cong 4$

Instrument example: powerset

- Take a predicate $p \colon X \to \mathcal{P}(2) \cong 4$
- Then $\mathrm{instr}_p \colon X \to \mathcal{P}(X + X)$ is:

$$\mathrm{instr}_p(x) = \{\kappa_1 x \mid 1 \in p(x)\} \,\cup\, \{\kappa_2 x \mid 0 \in p(x)\}$$

Instrument example: powerset

- Take a predicate $p\colon X \to \mathcal{P}(2) \cong 4$
- Then $\mathrm{instr}_p\colon X \to \mathcal{P}(X+X)$ is:

$$\mathrm{instr}_p(x) = \{\kappa_1 x \mid 1 \in p(x)\} \;\cup\; \{\kappa_2 x \mid 0 \in p(x)\}$$

- These instruments are not side-effect-free:

$$\big(\nabla \bullet \mathrm{instr}_p\big)(x)$$

Instrument example: powerset

- Take a predicate $p\colon X \to \mathcal{P}(2) \cong 4$
- Then $\mathrm{instr}_p\colon X \to \mathcal{P}(X+X)$ is:

$$\mathrm{instr}_p(x) = \{\kappa_1 x \mid 1 \in p(x)\} \;\cup\; \{\kappa_2 x \mid 0 \in p(x)\}$$

- These instruments are not side-effect-free:

$$\big(\nabla \bullet \mathrm{instr}_p\big)(x) \;=\; \{x \mid 1 \in p(x) \text{ or } 0 \in p(x)\}$$

Instrument example: powerset

- Take a predicate $p\colon X \to \mathcal{P}(2) \cong 4$
- Then $\mathrm{instr}_p\colon X \to \mathcal{P}(X+X)$ is:

$$\mathrm{instr}_p(x) = \{\kappa_1 x \mid 1 \in p(x)\} \;\cup\; \{\kappa_2 x \mid 0 \in p(x)\}$$

- These instruments are not side-effect-free:

$$(\nabla \bullet \mathrm{instr}_p)(x) \;=\; \{x \mid 1 \in p(x) \text{ or } 0 \in p(x)\} \;=\; \begin{cases} \{x\} & \text{if } p(x) \neq \emptyset \\ \emptyset & \text{if } p(x) = \emptyset. \end{cases}$$

Instrument example: powerset

- Take a predicate $p\colon X \to \mathcal{P}(2) \cong 4$
- Then $\mathrm{instr}_p\colon X \to \mathcal{P}(X+X)$ is:

$$\mathrm{instr}_p(x) = \{\kappa_1 x \mid 1 \in p(x)\} \,\cup\, \{\kappa_2 x \mid 0 \in p(x)\}$$

- These instruments are not side-effect-free:

$$(\nabla \bullet \mathrm{instr}_p)(x) = \{x \mid 1 \in p(x) \text{ or } 0 \in p(x)\} = \begin{cases} \{x\} & \text{if } p(x) \neq \emptyset \\ \emptyset & \text{if } p(x) = \emptyset. \end{cases}$$

- For the (affine) non-empty powerset the case $p(x) = \emptyset$ does not occur, so we get side-effect-freeness.

iCIS | Digital Security
Radboud University

Instrument example: state monad

Instrument example: state monad

- Consider $T(X) = (S \times X)^S$, for a fixed set of states S

Instrument example: state monad

- Consider $T(X) = (S \times X)^S$, for a fixed set of states S
- A predicate is a map $p\colon X \to (S + S)^S$

Instrument example: state monad

- Consider $T(X) = (S \times X)^S$, for a fixed set of states S
- A predicate is a map $p \colon X \to (S + S)^S$
- The associated instrument $\mathrm{instr}_p \colon X \to (S \times X + S \times X)^S$ is:

$$\mathrm{instr}_p(x)(s) = \begin{cases} \kappa_1(s', x) \text{ if } p(x)(s) = \kappa_1 s' \\ \kappa_2(s', x) \text{ if } p(x)(s) = \kappa_2 s' \end{cases}$$

Instrument example: state monad

- Consider $T(X) = (S \times X)^S$, for a fixed set of states S
- A predicate is a map $p \colon X \to (S + S)^S$
- The associated instrument $\mathrm{instr}_p \colon X \to (S \times X + S \times X)^S$ is:

$$\mathrm{instr}_p(x)(s) = \begin{cases} \kappa_1(s', x) \text{ if } p(x)(s) = \kappa_1 s' \\ \kappa_2(s', x) \text{ if } p(x)(s) = \kappa_2 s' \end{cases}$$

This instrument incorporates the side-effects of the predicate p

iCIS | Digital Security
Radboud University

Intermezzo: quantum instruments

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:

$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} + \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$

iCIS | Digital Security
Radboud University

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:
$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$
- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:
$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \;+\; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$
- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:

$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$

- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:

$$\big(\mathrm{instr}_p \circ \Delta\big)(x)$$

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:
$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} + \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$
- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:
$$\big(\mathrm{instr}_p \circ \Delta\big)(x) = \mathrm{instr}_p(x, x)$$

iCIS | Digital Security
Radboud University

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:

$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$

- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:

$$\begin{aligned} \big(\mathrm{instr}_p \circ \Delta\big)(x) &= \mathrm{instr}_p(x, x) \\ &= \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot x \cdot \sqrt{1-p} \end{aligned}$$

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:

$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$

- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:

$$\begin{aligned}(\mathrm{instr}_p \circ \Delta)(x) &= \mathrm{instr}_p(x, x) \\ &= \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot x \cdot \sqrt{1-p} \\ &= \sqrt{p} \cdot \sqrt{p} \cdot x \; + \; \sqrt{1-p} \cdot \sqrt{1-p} \cdot x\end{aligned}$$

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:

$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \;+\; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$

- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:

$$\begin{aligned}
\big(\mathrm{instr}_p \circ \Delta\big)(x) &= \mathrm{instr}_p(x, x) \\
&= \sqrt{p} \cdot x \cdot \sqrt{p} \;+\; \sqrt{1-p} \cdot x \cdot \sqrt{1-p} \\
&= \sqrt{p} \cdot \sqrt{p} \cdot x \;+\; \sqrt{1-p} \cdot \sqrt{1-p} \cdot x \\
&= p \cdot x + (1-p) \cdot x
\end{aligned}$$

Intermezzo: quantum instruments

- In the quantum model $\mathbf{vNA}^{\mathrm{op}}$ everything is turned around
- A predicate in a von Neumann algebra A is a $p \in A$ with $0 \leq p \leq 1$
- The associated instrument is a function $\mathrm{instr}_p \colon A \oplus A \to A$, given by:

$$\mathrm{instr}_p(x, y) = \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot y \cdot \sqrt{1-p}.$$

- Side-effect-freeness means $\mathrm{instr}_p \circ \Delta = \mathrm{id}$
- **Important:** commutative vNA's are side-effect-free:

$$\begin{aligned}
\big(\mathrm{instr}_p \circ \Delta\big)(x) &= \mathrm{instr}_p(x, x) \\
&= \sqrt{p} \cdot x \cdot \sqrt{p} \; + \; \sqrt{1-p} \cdot x \cdot \sqrt{1-p} \\
&= \sqrt{p} \cdot \sqrt{p} \cdot x \; + \; \sqrt{1-p} \cdot \sqrt{1-p} \cdot x \\
&= p \cdot x + (1-p) \cdot x \\
&= x.
\end{aligned}$$

Where we are, sofar

Strong affineness

Strong affineness

- If T is affine, then predicates give side-effect-free instruments

iCIS | Digital Security
Radboud University

Strong affineness

- If T is affine, then predicates give side-effect-free instruments
- For a bijective correspondence we need a stronger property

Strong affineness

- If T is affine, then predicates give side-effect-free instruments
- For a bijective correspondence we need a stronger property

Definition

A (strong) monad T is called strongly affine if the following squares are pullbacks

$$\begin{array}{ccc} T(X) \times Y & \xrightarrow{\pi_2} & Y \\ {\scriptstyle \mathrm{st}_1}\downarrow & & \downarrow{\scriptstyle \eta_Y} \\ T(X \times Y) & \xrightarrow[T(\pi_2)]{} & T(Y) \end{array}$$

iCIS | Digital Security
Radboud University

Strong affineness

- If T is affine, then predicates give side-effect-free instruments
- For a bijective correspondence we need a stronger property

Definition

A (strong) monad T is called strongly affine if the following squares are pullbacks

$$\begin{array}{ccc} T(X) \times Y & \xrightarrow{\pi_2} & Y \\ \downarrow{\scriptstyle \mathrm{st}_1} & & \downarrow{\scriptstyle \eta_Y} \\ T(X \times Y) & \xrightarrow[T(\pi_2)]{} & T(Y) \end{array}$$

(Strongly affine implies affine)

iCIS | Digital Security
Radboud University

Strongly affine (counter)examples

Strongly affine (counter)examples

- The standard affine monad examples $\mathcal{P}_+$, $\mathcal{D}$, $\mathcal{G}$, $\mathcal{E}$ and $\mathcal{R}$ are also strongly affine
  - (proofs are not entirely trivial)

Strongly affine (counter)examples

- The standard affine monad examples $\mathcal{P}_+$, $\mathcal{D}$, $\mathcal{G}$, $\mathcal{E}$ and $\mathcal{R}$ are also strongly affine
  - (proofs are not entirely trivial)
- (Kenta Cho) The monad $\mathcal{D}_\pm$ is affine but not strongly affine

Strongly affine (counter)examples

- The standard affine monad examples $\mathcal{P}_+$, $\mathcal{D}$, $\mathcal{G}$, $\mathcal{E}$ and $\mathcal{R}$ are also strongly affine
  - (proofs are not entirely trivial)
- (Kenta Cho) The monad $\mathcal{D}_\pm$ is affine but not strongly affine

Strongly affine (counter)examples

- The standard affine monad examples $\mathcal{P}_+$, $\mathcal{D}$, $\mathcal{G}$, $\mathcal{E}$ and $\mathcal{R}$ are also strongly affine
  - (proofs are not entirely trivial)
- (Kenta Cho) The monad $\mathcal{D}_\pm$ is affine but not strongly affine
  - $\mathcal{D}_\pm(X)$ contains $\sum_i r_i | x_i \rangle$ with $r_i \in \mathbb{R}$ and $\sum_i r_i = 1$
  - In this monad $\mathcal{D}_\pm$ there is interference: positive and negative factors can cancel each other out

Strongly affine monads and instruments

Strongly affine monads and instruments

Theorem (I)

If T is strongly affine, then there is a bijective corrrespondence

$$\frac{\textit{predicates}}{\overline{\overline{\textit{side-effect-free instruments}}}}$$

Strongly affine monads and instruments

Theorem (I)

If T is strongly affine, then there is a bijective corrrespondence

$$\frac{\textit{predicates}}{\overline{\overline{\textit{side-effect-free instruments}}}}$$

More precisely, the correspondence is between maps in $\mathcal{K}\ell(T)$,

$$\frac{X \xrightarrow{\;p\;} 2}{\overline{\overline{X \xrightarrow[f]{} X + X \quad \textit{with } \nabla \bullet f = id}}}$$

Relating commutativity

Relating commutativity

Theorem (II)

If the monad T is commutative, then instruments commute — giving commutativity in an effectus-theoretic sense.

Relating commutativity

Theorem (II)

If the monad T is commutative, then instruments commute — giving commutativity in an effectus-theoretic sense.

More precisely, for predicates $p, q \colon X \to 2$ we have:

$$\begin{array}{ccccc} X & \xrightarrow{instr_{p}} & X+X & \xrightarrow{q+q} & 2+2 \\ \| & & & & \cong \big\downarrow [\kappa_1+\kappa_1, \kappa_2+\kappa_2] \\ X & \xrightarrow[instr_{q}]{} & X+X & \xrightarrow[p+p]{} & 2+2 \end{array}$$

Relating commutativity

Theorem (II)

If the monad T is commutative, then instruments commute — giving commutativity in an effectus-theoretic sense.

More precisely, for predicates $p, q \colon X \to 2$ we have:

$$
\begin{array}{ccccc}
X & \xrightarrow{\;instr_p\;} & X + X & \xrightarrow{\;q+q\;} & 2 + 2 \\
\| & & & & \cong \big\downarrow [\kappa_1+\kappa_1, \kappa_2+\kappa_2] \\
X & \xrightarrow[\;instr_q\;]{} & X + X & \xrightarrow[\;p+p\;]{} & 2 + 2
\end{array}
$$

The isomorphism on the right can be illustrated as:

$$
\begin{array}{ccccccccccc}
2+2 & = & (1 & + & 1) & + & (1 & + & 1) \\
 & & \downarrow & & \searrow & & \swarrow & & \downarrow \\
2+2 & = & (1 & + & 1) & + & (1 & + & 1)
\end{array}
$$

Where we are, sofar

Final remarks

Final remarks

- Quantum theory forms a rich source of inspiration for program semantics and logic — and for coalgebra in particular
- Recent formalisation in terms of effectuses
    - framework deals with side-effects of observations
    - Boolean and probabilitistic computation given by subclasses
- Characterising the commutative (probabilistic and side-effect-free) fragment is an open challenge
    - Kleisli categories of suitable monads play an important role
- This CMCS paper clarifies the role of strong affiness and of commutativity of the monad