

Trace models of concurrent valuation algebras

Nasos Evangelou-Oost
& Callum Bannister, Larissa Meinicke, Ian J. Hayes

2 June 2023



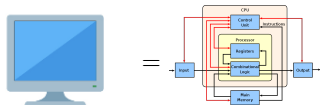
- Concurrent & distributed systems are everywhere, yet really hard to reason about.
- Compositional methods mitigate the combinatorial explosion of possibilities when many interacting systems combine.
- Concurrent valuation algebras (CVAs) are . . .
 - compositional, modular, local,
 - algebraic, lattice-based,
 - able to detect subtle information inconsistency (e.g. sequential inconsistency),
 - algorithmically tractable.

- 1 OVAs, CVAs
- 2 Trace models
- 3 Morphisms
- 4 Local computation
- 5 Summary & future work

Topologies and presheaves

Topologies encode the connective structure of distributed and concurrent systems:

- Topology \mathcal{T} for distributed systems: networks.
- Topology \mathcal{T} for concurrent systems: shared memory and resources.



A **presheaf** is a contravariant functor on a topology \mathcal{T}

$$\Phi : \mathcal{T}^{op} \rightarrow \mathcal{P}os$$

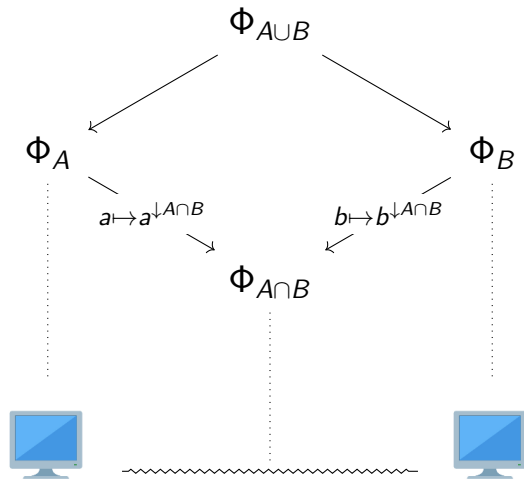
I.e. we have,

- **Object map**: A function sending each domain $A \in \mathcal{T}$ to a poset Φ_A .
- **Arrow map**: A function sending each inclusion $B \subseteq A$ to a monotone **restriction map** $a \mapsto a^{\downarrow B} : \Phi_A \rightarrow \Phi_B$.
- **Transitivity**: For $C \subseteq B \subseteq A$ and $a \in \Phi_A$,

$$(a^{\downarrow B})^{\downarrow C} = a^{\downarrow C}$$

- **Identity law**: For each $A \in \mathcal{T}$ and $a \in \Phi_A$,

$$a^{\downarrow A} = a$$



Given $\Phi : \mathcal{T}^{op} \rightarrow \mathcal{Pos}$, combine all the posets Φ_A into one big poset $\int \Phi$.

$$\int \Phi = \{(A, a) \mid A \in \mathcal{T}, a \in \Phi_A\}$$

Partial order \preceq :

$$(A, a) \preceq (B, b) \iff B \subseteq A \text{ and } a^{\downarrow B} \leq_{\Phi_B} b$$

Shorthand: write a instead of (A, a) leaving domain implicit; use **domain** map d to recover domain

$$\begin{aligned} d : \int \Phi &\rightarrow \mathcal{T}^{op} \\ da &= \text{domain of } a \end{aligned}$$

Ordered valuation algebras (OVAs)

Definition. An **ordered valuation algebra (OVA)** consists of...

- a poset-valued presheaf

$$\Phi : \mathcal{T}^{op} \rightarrow \mathcal{P}_{\mathcal{O}\mathcal{S}}$$

- a binary operator

$$\otimes : \int \Phi \times \int \Phi \rightarrow \int \Phi$$

- a natural transformation

$$\epsilon : \mathbf{1} \Rightarrow \Phi$$

(i.e. an element $\epsilon_A \in \Phi_A$ for each A such that $\epsilon_A^{\downarrow B} = \epsilon_B$ whenever $B \subseteq A$.)

Obeying the following axioms...

- **Ordered semigroup:**

\otimes is associative and monotone

- **Labelling:**

$$d(a \otimes b) = da \cup db$$

- **Neutrality:**

$$\epsilon_{da} \otimes a = a = a \otimes \epsilon_{da}$$

- **Combination:**

$$(a \otimes b)^{\downarrow da} = a \otimes b^{\downarrow da \cap db} \text{ and } (a \otimes b)^{\downarrow db} = a^{\downarrow da \cap db} \otimes b$$

Theorem. Let $(\Phi, \otimes, \epsilon)$ be an OVA. For each inclusion $B \subseteq A$ in \mathcal{T} , there is an adjunction

$$a^{\downarrow B} \leq_{\Phi_B} b \iff a \leq_{\Phi_A} \epsilon_a \otimes b$$

We call $b \mapsto \epsilon_a \otimes b$ **extension of b to A** and write $b^{\uparrow A} := \epsilon_a \otimes b$.

Some facts:

- Extension is a functor: if $C \subseteq B \subseteq A$ and $c \in \Phi_C$, then

$$(c^{\uparrow B})^{\uparrow A} = c^{\uparrow A} \text{ and } c^{\uparrow C} = c$$

- If Φ has the **strongly neutral property**: $\forall A, B \in \mathcal{T}$ we have $\epsilon_A \otimes \epsilon_B = \epsilon_{A \cup B}$, then for all $B \subseteq A$,

$$\epsilon_B^{\uparrow A} = \epsilon_A$$

- Restriction after extension is the identity map, i.e., for $B \subseteq A$ and $b \in \Phi_B$,

$$(b^{\uparrow A})^{\downarrow B} = b$$

- Extension after restriction is a **closure operator**, i.e., for $B \subseteq A$ and $a \in \Phi_A$,

$$a \leq_{\Phi_A} (a^{\downarrow B})^{\uparrow A} \quad \text{and} \quad (((a^{\downarrow B})^{\uparrow A})^{\downarrow B})^{\uparrow A} = (a^{\downarrow B})^{\uparrow A}$$

- If for each $A \in \mathcal{A}$, Φ_A is a complete lattice, then so is $\int \Phi$.

Definition. A *concurrent valuation algebra (CVA)* consists of...

- A commutative OVA

$$(\Phi, \parallel, \delta)$$

- A (generally noncommutative) OVA

$$(\Phi, \circ, \epsilon)$$

Obeying the following axioms...

- *Weak exchange:*

$$(a \parallel b) \circ (c \parallel d) \preceq (a \circ c) \parallel (b \circ d)$$

- *Neutral laws:*

$$\epsilon_A \preceq \epsilon_A \parallel \epsilon_A \text{ and } \delta_A \circ \delta_A \preceq \delta_A$$

- *Refinement:*

$$a \preceq b \iff db \subseteq da \text{ and } a^{\downarrow db} \leq_{\Phi_{db}} b$$

- *Hoare triple (Hoare logic):*

$$p \{a\} q := p \circ a \preceq q$$

- *Jones quintuple (rely-guarantee logic):*

$$p \ r \ \{a\} \ g \ q := p \ \{a \parallel r\} \ q \text{ and } a \preceq g^{-1}$$

¹Ian doesn't like this.

1 OVAs, CVAs

2 Trace models

3 Morphisms

4 Local computation

5 Summary & future work

- Trace models: specifications are subsets of *traces* = lists of observations $t = [t_1, \dots, t_n]$.
- Can be coupled to a clock or not. What does the i in t_i mean?
- Decoupling from a global clock = stuttering invariance (traces of states);

$$[1, 1, 2, 3, 3, 3] \sim [1, 2, 2, 3]$$

Two ways to encode stuttering-invariance:

- Stuttering closure: if the first trace belongs to a specification, so does the second.
- Stuttering reduction: both traces are identified in a quotient.

Depends on what the elements t_i in a trace $t = [t_1, \dots, t_n]$ mean:

- **State:** t_i is a state of the system.

- Parallel = sync = $\wedge_A = \cap$,

$$\{[a, b, c]\} \wedge_A \{[a, b, c]\} = \{[a, b, c]\}$$

- Sequential = gluing concatenation = \smile .

$$\{[a, b, c]\} \smile_A \{[c, d]\} = \{[a, b, c, d]\}$$

- **Action:** t_i is an action taken by the system.

- Parallel = shuffle = \sqcup ,

$$\{[a, b, c]\} \sqcup_A \{[d, e]\} = \{[a, b, d, e, c], [d, a, b, c, e], [a, b, d, c, e], \dots\}$$

- Sequential = concatenation = \frown ,

$$\{[a, b, c]\} \frown_A \{[d, e]\} = \{[a, b, c, d, e]\}$$

State trace CVA Σ

- State functor: $\Omega_A^{\text{state}} := A \rightarrow \mathbb{Z}$ (A -tuples)
- Nonempty list functor: $L_+ : \text{Set} \rightarrow \text{Set}$
- Powerset functor: $P : \text{Set} \rightarrow \mathcal{P}\text{os}$

State trace CVA Σ is defined as the composite,

$$\Sigma = \mathcal{T}^{op} \xrightarrow{\Omega^{\text{state}}} \text{Set} \xrightarrow{L_+} \text{Set} \xrightarrow{P} \mathcal{P}\text{os}$$

- Parallel = synchronisation = meet:

$$\begin{aligned} \wedge : \int \Sigma \times \int \Sigma &\rightarrow \int \Sigma \\ a \wedge b &:= a^{\uparrow \text{da} \cup \text{db}} \cap b^{\uparrow \text{da} \cup \text{db}} \end{aligned}$$

- Sequential = gluing concatenation, matches final/initial states:

$$\begin{aligned} \smile : \int \Sigma \times \int \Sigma &\rightarrow \int \Sigma \\ a \smile b &:= a^{\uparrow \text{da} \cup \text{db}} \smile_{\text{da} \cup \text{db}} b^{\uparrow \text{da} \cup \text{db}} \end{aligned}$$

- Neutral elements for parallel, sequential resp.

$$\begin{aligned} \top : 1 &\Rightarrow \Sigma \\ \top_A &= L_+(\Omega_A^{\text{state}}) \end{aligned}$$

$$\begin{aligned} \nu : 1 &\Rightarrow \Sigma \\ \nu_A &= \{t \in L_+(\Omega_A^{\text{state}}) \mid \lambda(t) = 1\} \end{aligned}$$

- Action functor: $\Omega^{\text{act}} : \mathcal{T}^{op} \rightarrow \mathcal{Set}$
 - $\Omega_A^{\text{act}} = \mathbb{Z}^A \times \mathbb{Z}^A$ (events = pairs of states)
 - $\Omega_A^{\text{act}} = P(\mathbb{Z}^A \times \mathbb{Z})$ (events with external choice)
 - $\Omega_A^{\text{act}} = A \times A \rightarrow \mathbb{Z}$ (linear functions $\mathbb{Z}^A \rightarrow \mathbb{Z}^A$)

Action trace CVA Γ is defined as the composite:

$$\Gamma = \mathcal{T}^{op} \xrightarrow{\Omega^{\text{act}}} \mathcal{Set} \xrightarrow{L} \mathcal{Set} \xrightarrow{P} \mathcal{Pos}$$

- Parallel = shuffle:

$$\begin{aligned} \sqcup : \int \Gamma \times \int \Gamma &\rightarrow \int \Gamma \\ a \sqcup b &:= a^{\uparrow_{\text{da} \cup \text{db}}} \sqcup_{\text{da} \cup \text{db}} b^{\uparrow_{\text{da} \cup \text{db}}} \end{aligned}$$

- Sequential = concatenation:

$$\begin{aligned} \frown : \int \Gamma \times \int \Gamma &\rightarrow \int \Gamma \\ a \frown b &:= a^{\uparrow_{\text{da} \cup \text{db}}} \frown_{\text{da} \cup \text{db}} b^{\uparrow_{\text{da} \cup \text{db}}} \end{aligned}$$

- Neutral element (same for \sqcup and \frown):

$$\begin{aligned} \iota : 1 &\Rightarrow \Gamma \\ \iota_A &= \{[\]_A\} \end{aligned}$$

- **Free semigroup with idempotent generators** $I : \text{Set} \rightarrow \text{Semi}$
 $I(S) = L_+(S)$ modulo $ss \sim s$ for all $s \in S$, with concatenation as multiplication.

Relative state trace CVA Σ^{rel} is defined as the composite:

$$\Sigma^{\text{rel}} = \mathcal{T}^{\text{op}} \xrightarrow{\Omega^{\text{state}}} \text{Set} \xrightarrow{I} \text{Semi} \xrightarrow{U} \text{Set} \xrightarrow{P} \mathcal{Pos}$$

Combine operators and neutral elements defined as in Σ .

- Projection and parallel no longer preserve the length of traces.
 Parallel syncs on common variables, shuffles on disjoint variables. E.g. $a, b \in \int \Sigma^{\text{rel}}$ with $\text{da} = \{x, y\}$ and $\text{db} = \{y, z\}$,

$$\begin{aligned} a \wedge^{\text{rel}} b &= \left\{ \begin{bmatrix} x_0 & \boxed{x_1} & \boxed{x_1} \\ y_0 & y_0 & \boxed{y_1} \end{bmatrix} \right\} \wedge \left\{ \begin{bmatrix} y_0 & y_0 & \boxed{y_1} \\ z_0 & \boxed{z_1} & \boxed{z_1} \end{bmatrix} \right\} \\ &= \left\{ \begin{bmatrix} x_0 & \boxed{x_1} & \boxed{x_1} \\ y_0 & y_0 & \boxed{y_1} \\ z_0 & \boxed{z_1} & \boxed{z_1} \end{bmatrix}, \begin{bmatrix} x_0 & \boxed{x_1} & \boxed{x_1} & \boxed{x_1} \\ y_0 & y_0 & y_0 & \boxed{y_1} \\ z_0 & z_0 & \boxed{z_1} & \boxed{z_1} \end{bmatrix}, \begin{bmatrix} x_0 & x_0 & \boxed{x_1} & \boxed{x_1} \\ y_0 & y_0 & y_0 & \boxed{y_1} \\ z_0 & \boxed{z_1} & \boxed{z_1} & \boxed{z_1} \end{bmatrix} \right\} \end{aligned}$$

1 OVAs, CVAs

2 Trace models

3 **Morphisms**

4 Local computation

5 Summary & future work

Definition. A **lax morphism of OVAs** $f : (\Phi, \otimes, \epsilon) \rightarrow (\Phi', \otimes', \epsilon')$ is a family of maps $\{f_A : \Phi_A \rightarrow \Phi'_A\}_{A \in \mathcal{T}}$, obeying for all $a \in \Phi_A$, $b \in \Phi_B$ and $C \subseteq A \dots$

- **Monotonicity:**

$$a \preceq b \implies f_A(a) \preceq f_B(b)$$

- **Lax naturality:**

$$f_A(a)^{\downarrow C} \preceq f_C(a^{\downarrow C})$$

- **Lax multiplicativity.**

$$f_A(a) \otimes' f_B(b) \preceq f_{A \cup B}(a \otimes b)$$

- **Lax unitality.**

$$\epsilon'_A \preceq f_A(\epsilon_A)$$

Definition. A **lax morphism of CVAs** $f : (\Phi, \parallel, \delta, \circ, \epsilon) \rightarrow (\Phi', \parallel', \delta', \circ', \epsilon')$ is a map f that is both...

- a lax morphism of (commutative) OVAs $(\Phi, \parallel, \delta) \rightarrow (\Phi', \parallel', \delta)$,
- a lax morphism of OVAs $(\Phi, \circ, \epsilon) \rightarrow (\Phi', \circ', \epsilon)$.

Reversing the inequalities above defines a **colax morphism**.

Colax morphism from Σ to Σ^{rel}

Notice $L_+ \cong U \circ F$ where F is the free semigroup functor.

For each set X there is a quotient map $q_X : F(X) \twoheadrightarrow I(X) \cong F(X)/\sim$ where \sim is the congruence generated by $x = xx$ for all $x \in X$. This map does a *stuttering reduction*.

This is natural in X , so we get a natural transformation $q : F \Rightarrow I$.

By composing (whiskering), this gives a natural transformation $f : \Sigma \Rightarrow \Sigma^{\text{rel}}$.

$$f = \mathcal{T}^{op} \xrightarrow{\Omega^{\text{state}}} \text{Set} \begin{array}{c} \xrightarrow{F} \\ \Downarrow q \\ \xrightarrow{I} \end{array} \text{Semi} \xrightarrow{U} \text{Set} \xrightarrow{P} \mathcal{Pos}$$

Moreover, f is a colax morphism of CVAs:

- $a \preceq b \implies f(a) \preceq f(b)$
- $f(\top) \preceq \top^{\text{rel}}$
- $f(\nu) \preceq \nu^{\text{rel}}$
- $f(a \wedge b) \preceq f(a) \wedge^{\text{rel}} f(b)$ (note this is not equality: consider $a = [xx]$ and $b = [x]$.)
- $f(a \smile b) \preceq f(a) \smile^{\text{rel}} f(b)$

There should be a right-adjoint $g : \Sigma^{\text{rel}} \rightarrow \Sigma$ that does stuttering closure...

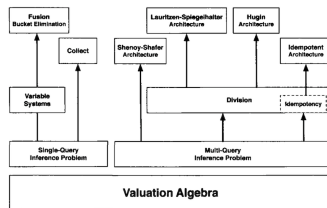
- 1 OVAs, CVAs
- 2 Trace models
- 3 Morphisms
- 4 Local computation**
- 5 Summary & future work

Efficient computation is enabled by the *combination* law:

$$(a \otimes b)^{\downarrow da} = a \otimes b^{\downarrow da \cap db}$$

$$(a \otimes b)^{\downarrow db} = a^{\downarrow da \cap db} \otimes b$$

The LHS is usually *expensive* to compute, but the RHS can be *much cheaper* if $da \cap db$ is small.



- Local computation framework specialises to standard optimal algorithms in databases, probabilistic graphical models, CSPs, dynamic programming, ...
- We can already use these algorithms in a CVA on just parallel combinations.
- Need to generalise to noncommutative case & with 2 combine operators.
- Theoretically feasible from the generalised combination rule, but not yet implemented.

- 1 OVAs, CVAs
- 2 Trace models
- 3 Morphisms
- 4 Local computation
- 5 Summary & future work

Summary & future work

Summary:

- We introduced CVAs, a modular/local/compositional algebra for specifying concurrent/distributed systems.
- CVAs support a duoidal parallel/sequential algebraic structure, refinement, Hoare and rely-guarantee logics.
- We described 3 example trace models of CVAs and related them by morphisms.
- CVAs have a path to efficient algorithms for inference, static analysis, model checking. . .

Future work:

- More models: Aczel trace models, graph models, sheaf based models that better capture local properties. . .
- Push-forward/pullback of OVAs/CVAs on different spaces.
- Develop local computation algorithms.
- Formalisation in Isabelle.

References:



Evangelou-Oost, Nasos, Callum Bannister and Ian J. Hayes (2023). 'Contextuality in Distributed Systems'. In: *Relational and Algebraic Methods in Computer Science*. Springer International Publishing, pp. 52–68.



Evangelou-Oost, Nasos, Callum Bannister, Larissa Meinicke et al. (2023). *Trace models of concurrent valuation algebras*. arXiv: 2305.18017 [cs.LO].

Contact: naso@oneironaut.dev, linkedin.com/in/nasosev

THANK YOU!