Trace models of concurrent valuation algebras

Naso Evangelou-Oost[1] Larissa Meinicke[1] Callum Bannister[1] Ian J. Hayes[1]

[1]School of Electrical Engineering and Computer Science,
The University of Queensland

The first author is supported by the Australian Government Research Training Program Scholarship

ICFEM23
2023–11–22

THE UNIVERSITY OF QUEENSLAND
AUSTRALIA

Introduction

- **Challenges in Reasoning:** Concurrent and distributed systems are prevalent but complex to analyse.
- **Compositional Methods:** These methods address the exponential growth of possibilities in interacting systems.
- **Concurrent Valuation Algebras (CVAs):**
  - *Attributes:* Compositional, modular, local, algebraic, lattice-based.
  - *Generalisation:* Generalises models like CSP (Communicating Sequential Processes), CKA (Concurrent Kleene Algebra), CRA (Concurrent Refinement Algebra).
  - *Advantages:* Detects subtle information inconsistencies (RAMiCS23 paper), supports Hoare and rely-guarantee logics.

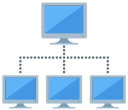
Project Goal:

- ***Developing a Localised Refinement Algebra*** for modular and compositional specifications in concurrent/distributed systems, generalising existing models, and supporting standard program logics.

Topologies and Prealgebras

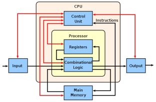
Encoding Connective Structures:

- **Distributed Systems:** Represented as networks.



- **Concurrent Systems:** Characterised by shared memory and resources.

$=$



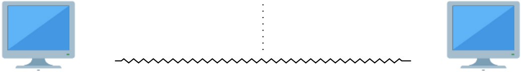
Prealgebra: A poset-(partially ordered set) valued contravariant functor on a topology $\mathscr{T}$ that captures the connective structure of a concurrent/distributed system.

$$\Phi : \mathscr{T}^{op} \to \mathscr{P}os$$

- **Object Map:** Maps each domain $A \in \mathscr{T}$ to a poset Φ_A. ***(Φ_A is the poset of specifications on A.)***
- **Arrow Map:** Maps each inclusion $B \subseteq A$ to a monotone restriction map $a \mapsto a^{\downarrow B} : \Phi_A \to \Phi_B$. ***(Focuses a specification $a \in \Phi_A$ to a subdomain $B \subseteq A$, yielding a specification $a^{\downarrow B}$.)***
- **Transitivity & Identity Law:** For $C \subseteq B \subseteq A$, $(a^{\downarrow B})^{\downarrow C} = a^{\downarrow C}$; for each A and $a \in \Phi_A$, $a^{\downarrow A} = a$.

Prealgebra on a distributed system

$$
\begin{tikzcd}
 & \Phi_{A\cup B} \arrow[dl, "c \mapsto c^{\downarrow A}"'] \arrow[dr, "c \mapsto c^{\downarrow B}"] & \\
\Phi_A \arrow[dr, "a \mapsto a^{\downarrow A\cap B}"'] & & \Phi_B \arrow[dl, "b \mapsto b^{\downarrow A\cap B}"] \\
 & \Phi_{A\cap B} &
\end{tikzcd}
$$



(Covariant) Grothendieck Construction

Combining Posets: Given a prealgebra $\Phi : \mathcal{T}^{op} \to \mathcal{P}os$, the Grothendieck construction combines all posets Φ_A into a single poset $\int \Phi$.

Definition of $\int \Phi$:

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

Partial 'Grothendieck' Order $\preceq$:

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

Shorthand Notation:

- Write a instead of (A, a), with the domain being implicit.
- Use the ***domain*** map d to recover the domain of an element:

$$\mathrm{d} : \int \Phi \to \mathcal{T}^{op}$$
$$\mathrm{d}a = \mathrm{d}(A, a) = A$$

Definition: An ***ordered valuation algebra (OVA)*** comprises:

- A ***prealgebra***:

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

- A ***binary operator***:

$$\otimes : \int \Phi \times \int \Phi \to \int \Phi$$

- A ***neutral element*** ϵ, comprising an element $\epsilon_A \in \Phi_A$ for each $A \in \mathcal{T}$, such that $\epsilon_A^{\downarrow B} = \epsilon_B$ for all $B \subseteq A$.

OVA Axioms:

- ***Ordered Semigroup:*** $\otimes$ is associative and monotone.
- ***Labelling:***

$$\mathrm{d}(a \otimes b) = \mathrm{d}a \cup \mathrm{d}b$$

- ***Neutrality:***

$$\epsilon_{\mathrm{d}a} \otimes a = a = a \otimes \epsilon_{\mathrm{d}a}$$

- ***Combination:***

$$(a \otimes b)^{\downarrow \mathrm{d}a} = a \otimes b^{\downarrow \mathrm{d}a \cap \mathrm{d}b} \text{ and } (a \otimes b)^{\downarrow \mathrm{d}b} = a^{\downarrow \mathrm{d}a \cap \mathrm{d}b} \otimes b$$

Extension in Ordered Valuation Algebras

Theorem: For an OVA $(\Phi, \otimes, \epsilon)$, for each inclusion $B \subseteq A$ in $\mathcal{T}$, there is a Galois connection: $\forall\ a \in \Phi_A,\ b \in \Phi_B$,

$$a^{\downarrow B} \leq_{\Phi_B} b \iff a \leq_{\Phi_A} \epsilon_{\mathrm{d}a} \otimes b$$

Define the ***extension of*** b ***to*** A as

$$b^{\uparrow A} := \epsilon_{\mathrm{d}a} \otimes b$$

Interpretation: Extension allows lifting a specification from a domain B to a larger domain A by permitting any behaviour outside B.

Key Facts:

- **Functorial Nature of Extension:** If $C \subseteq B \subseteq A$, $(c^{\uparrow B})^{\uparrow A} = c^{\uparrow A}$ and $c^{\uparrow C} = c$.
- **Restriction-Extension Identity:** $(b^{\uparrow A})^{\downarrow B} = b$.
- **Closure Properties:** Extension after restriction is a closure operator, ensuring $a \leq_{\Phi_A} (a^{\downarrow B})^{\uparrow A}$.
- **Lattice Completeness:** If each Φ_A is a complete lattice, $\int \Phi$ is also complete.

Concurrent Valuation Algebras (CVAs)

Definition: A ***concurrent valuation algebra (CVA)*** comprises:

- A ***commutative OVA***:

$$(\Phi, \|, \mathtt{run})$$

- A (usually noncommutative) ***OVA***:

$$(\Phi, \fatsemi, \mathtt{skip})$$

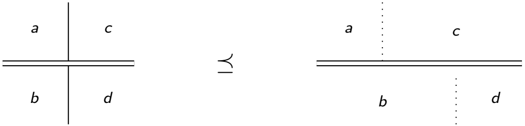
CVA Axioms:

- ***Weak Exchange:***

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

- ***Neutral Laws:***

$$\mathtt{skip}_A \preceq \mathtt{skip}_A \parallel \mathtt{skip}_A \text{ and } \mathtt{run}_A \fatsemi \mathtt{run}_A \preceq \mathtt{run}_A$$



Refinement & Reasoning in CVAs

- ***Refinement:***

$$a \preceq b \iff \mathrm{d}b \subseteq \mathrm{d}a \text{ and } a^{\downarrow \mathrm{d}b} \leq_{\Phi_{\mathrm{d}b}} b$$

- ***Hoare Triple (Hoare Logic):***

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

- ***Jones Quintuple (Rely-Guarantee Logic):***

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

Example (Concurrency Law): Let $p, p', a, a', q, q' \in \int \Phi$. Then:

$$p\ \{a\}\ q \text{ and } p'\ \left\{a'\right\}\ q' \implies (p \parallel p')\ \left\{a \parallel a'\right\}\ (q \parallel q')$$

Proof: Assume $p \fatsemi a \preceq q$ and $p' \fatsemi a' \preceq q'$. By weak exchange and monotonicity:

$$(p \parallel p') \fatsemi (a \parallel a') \preceq (p \fatsemi a) \parallel (p' \fatsemi a') \preceq q \parallel q'$$

Thus, $(p \parallel p')\ \{a \parallel a'\}\ (q \parallel q')$.

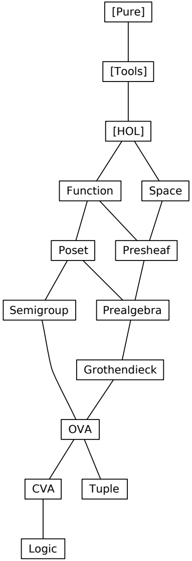
Models of Concurrent Valuation Algebras (CVAs)

- ***Action Trace Model:*** Traces as lists of atomic actions $t = [t_1, \ldots, t_n]$. Sequential composition as concatenation, parallel composition as interleaving.
- ***State Trace Model:*** Traces as nonempty lists of states $t = [t_1, \ldots, t_n]$. Sequential composition as gluing concatenation, parallel composition as synchronisation.
- ***Relative Trace Model:*** 'Stuttering-reduced' nonempty lists of states $t = [t_1, \ldots, t_n]$. Uses gluing concatenation and synchronisation for sequential and parallel compositions, respectively.
- ***CSP (Communicating Sequential Processes) Trace Model*:*** The domains of the topology are CSP alphabets.
- ***CKA (Concurrent Kleene Algebra) Algebraic Model*:*** The underlying topology is a one-point space.
- ***CRA (Concurrent Refinement Algebra) Algebraic Model*:*** Also based on a one-point space topology.

Computer Formalisation of CVAs

Summary:

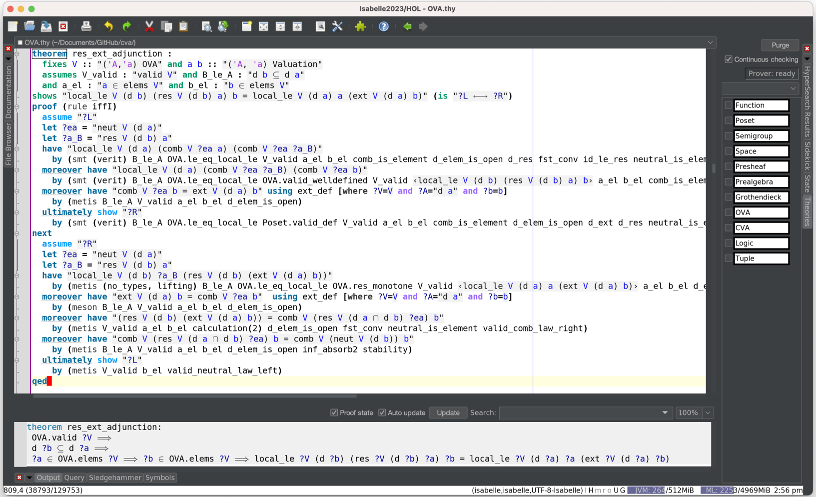
- **Formalisation in Isabelle/HOL:** Sections 2 and 3 of (ICFEM23 paper), covering abstract CVA theory, fully formalised.
- **Extension to Logics:** Includes formalisation of Hoare and Rely-Guarantee logics in CVAs, beyond the scope of the paper.
- **Open Source Collaboration:** The work is available at github.com/nasosev/cva. Collaborators are welcome :)



Theorem 1 Formalisation

Theorem (restriction-extension adjunction). Let $(\Phi, \otimes, \epsilon)$ be an OVA. Then for all $a, b \in \int \Phi$ with $\mathrm{d}b \subseteq \mathrm{d}a$,

$$a^{\downarrow \mathrm{d}b} \leq_{\Phi_{\mathrm{d}b}} b \iff a \leq_{\Phi_{\mathrm{d}a}} b^{\uparrow \mathrm{d}a}$$



Summary & Future Work

Summary:

- Introduced CVAs as a compositional algebra for concurrent/distributed systems.
- Highlighted CVAs' duoidal structure, refinement capabilities, and support for Hoare and rely-guarantee logics.
- Demonstrated multiple concrete models of CVAs relevant to concurrency paradigms.
- Indicated potential applications of local computation theory, with implications for static analysis and model checking.

Future Work: Exploration of further applications and theoretical extensions of CVAs.

References:

Evangelou-Oost, Naso et al. (2023). 'Trace Models of Concurrent Valuation Algebras'. In: ***Formal Methods and Software Engineering***. Ed. by Yi Li and Sofiène Tahar. Singapore: Springer Nature Singapore, pp. 118–136. ISBN: 978-981-99-7584-6.

Evangelou-Oost, Nasos, Callum Bannister and Ian J. Hayes (2023). 'Contextuality in Distributed Systems'. In: ***Relational and Algebraic Methods in Computer Science***. Ed. by Roland Glück, Luigi Santocanale and Michael Winter. Cham: Springer International Publishing, pp. 52–68. ISBN: 978-3-031-28083-2.

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

THANK YOU!