

Weird-Machine Reconstruction as Quantale-Enriched Semantics: Explanations Form Regular Languages and Are Never Unique

Third Projection Collective

Abela and Vella cast exploit analysis as reconstructing a weird machine (WM): a map $f : \Sigma \rightarrow L^*$ sending script fragments to strings of exploit-primitive labels, where each label l denotes a memory-state transition δ_l [1]. We give this a denotational semantics. (1) The definitions as stated do not compose: each δ_l has its own state abstraction $W \rightarrow W'$ and is called a function while being defined by constraints. Reading primitives as relations on a single product state space makes labels a monoid homomorphism $L^* \rightarrow \text{Rel}(W, W)$, i.e. a functor into the Kleisli category of the powerset monad. (2) The explanations $\text{Ex}(w, w') = \{s \in L^* : (w, w') \in \delta_s\}$ make W a category enriched in the quantale $\mathcal{P}(L^*)$ of languages, and reconstruction is computing its hom-objects. For finite abstractions each hom-object is regular. (3) For the paper's own allocation and free primitives on heap size we compute Ex exactly (checked by brute force on 12,495 cases). Every hom-object is infinite, and the string alloc-free explains every observation, so a label $f(p)$ is never determined by observed states; minimal explanations carry only the sign of the heap change. Open problems: a 2-categorical presentation of the equations between explanations, completeness of the stack-based compound-statement algorithm, and any re-validation of the original prototype, which we did not run.

1 The definitions, repaired

In [1], Definition 1 types a primitive's behaviour as $\delta_l : \mathcal{P}(W) \times \Sigma \rightarrow \mathcal{P}(W')$ over a primitive-specific abstraction, and §IV defines each δ_l by a constraint, e.g. $\delta_{\text{memalloc}}(\{w_0\}, p) = \{w'_0\}$ for all p with $w'_0 > w_0$. That constraint admits many w'_0 , so δ_l is a relation, not a function; and since $W \neq W'$ in general and each primitive uses a different W , the composite semantics of a label string l^* promised by Definition 2 is not defined. We fix both.

Definition 1. Let W be a set of abstract states (the product of all components used by any primitive). A primitive theory is a set L of labels with relations $\delta_l \subseteq W \times W$. Extend to strings by $\delta_\varepsilon = \text{id}_W$ and $\delta_{st} = \delta_s; \delta_t$ (relational composition). Given an observation (w, w') , its explanations are $\text{Ex}(w, w') = \{s \in L^* : (w, w') \in \delta_s\}$.

Components a primitive does not mention are unconstrained unless a frame condition is stated; [1] states one only for the code/stack region component ($w'_1 = w_1$). Dependence on the fragment p is dropped: no δ_l in [1] uses it.

Proposition 2. $s \mapsto \delta_s$ is the unique monoid homomorphism $L^* \rightarrow (\text{Rel}(W, W), ;, \text{id})$ extending $l \mapsto \delta_l$; equivalently a functor from the one-object free category on L into $\text{Kl}(\mathcal{P})$.

Proof. L^* is the free monoid on L and relational composition is associative with unit id_W ; $\text{Rel} = \text{Kl}(\mathcal{P})$. □

2 Explanations as an enriched category

$\mathcal{P}(L^*)$ with union and concatenation is a unital quantale (unit $\{\varepsilon\}$).

Theorem 3. Ex makes W a $\mathcal{P}(L^*)$ -enriched category: $\{\varepsilon\} \subseteq \text{Ex}(w, w)$ and $\text{Ex}(w, m) \cdot \text{Ex}(m, w') \subseteq \text{Ex}(w, w')$. Conversely, for any factorization $u = st$, $u \in \text{Ex}(w, w')$ iff $s \in \text{Ex}(w, m)$ and $t \in \text{Ex}(m, w')$ for some m .

Proof. $\delta_\varepsilon = \text{id}$ gives identities. If $(w, m) \in \delta_s$ and $(m, w') \in \delta_t$ then $(w, w') \in \delta_s; \delta_t = \delta_{st}$. Conversely if $(w, w') \in \delta_{st}$, by definition of relational composition some m has $(w, m) \in \delta_s$, $(m, w') \in \delta_t$. $\square$

Thus composition along observed intermediate states is sound for the compound-statement case of WMRecon (Algorithm 1 in [1]): labels found for consecutive sub-fragments concatenate to an explanation of the block. The converse direction, that WMRecon's stack discipline finds every explanation of a block, depends on which intermediate states the tool observes and is OPEN.

Proposition 4. If W is finite, every $\text{Ex}(w, w')$ is a regular language, recognized by the NFA with states W , transitions $x \xrightarrow{l} y$ for $(x, y) \in \delta_l$, start w , accept w' . The matrix $M = \sum_l \delta_l l$ over the Kleene algebra of languages has $\text{Ex} = M^*$.

Proof. A run of that NFA on s from w to w' is exactly a chain witnessing $(w, w') \in \delta_s$. M^* is the standard Kleene-algebra closure of a matrix [5]. $\square$

The NFA is the reconstructed WM read as an automaton; this matches Dullien's finite-state view of weird machines [4], which models the machine but not the analyst's explanation problem.

3 The allocation primitives, computed

Take the heap component $W = \mathbb{N}$ (allocated size) and $L = \{a, f\}$ with $\delta_a = \{(x, y) : y > x\}$, $\delta_f = \{(x, y) : y < x\}$, exactly as in [1, §IV-A].

Theorem 5. For $s \in \{a, f\}^*$ let $k_f(s)$ be the length of its leading run of f and $k_a(s)$ the length of its trailing run of a . Then $s \in \text{Ex}(w, w')$ iff: $s = \varepsilon$ and $w = w'$; or $s = a^k$, $k \geq 1$, $w' - w \geq k$; or $s = f^k$, $k \geq 1$, $w - w' \geq k$; or s contains both letters, $k_f(s) \leq w$ and $k_a(s) \leq w'$.

Proof. An a -run of length q from x can end at exactly the values $y \geq x + q$; an f -run of length r from x at exactly $0 \leq y \leq x - r$. Pure strings follow. Suppose both letters occur. Necessity: the leading f -run starts at w , so $w \geq k_f$; the trailing a -run starts at a value ≥ 0 , so $w' \geq k_a$. Sufficiency: send every f -run to 0 except a final f -run, which we send to w' ; send every a -run that is followed by an f -run of length r to $\max(x + q, r + w')$, which is legal and leaves room for that f -run. The leading f -run (if any) reaches 0 since $w \geq k_f$. If s ends in an a -run, it starts at 0 and reaches w' since $w' \geq k_a$. $\square$

`code/check.py` compares the theorem with brute-force relational composition for all $|s| \leq 7$ and $w, w' \leq 6$ (12,495 cases, intermediates bounded by 60): zero mismatches.

Corollary 6. (i) $af \in \text{Ex}(w, w')$ for all w, w' ; every hom-object is infinite. (ii) $\delta_{aa} \subsetneq \delta_a$, so labels are not idempotent and $\{a^k\}$ distinguishes counts only as lower bounds. (iii) The shortest explanation of (w, w') is ε , a or f according to the sign of $w' - w$.

Proof. (i) $k_f(af) = k_a(af) = 0$; and $af^n \in \text{Ex}(w, w)$ for all n . (ii) $(0, 1) \in \delta_a \setminus \delta_{aa}$. (iii) From Theorem 5. $\square$

So, for these primitives, the label $f(p)$ that WMRecon assigns is a choice and not a consequence of observations: the observation pair determines only a sign, and the five-allocation loop in [1, Fig. 2] is labelled “AllocPrimitive” consistently with this. Explanation counts grow quickly; for $(0, 3)$ the numbers of explanations of length $0, \dots, 7$ are $0, 1, 2, 4, 7, 15, 30, 60$ (computed). This is the WM analogue of proof multiplicity in e-graphs [3]. For the execute-crafted-code primitive with frame $w'_1 = w_1$, $\delta_e = \{((pc, C), (pc', C)) : pc' \notin C\}$ satisfies $\delta_e; \delta_e = \delta_e$ whenever some address lies outside C , so e^n for all $n \geq 1$ explain the same observations: an idempotent label.

4 Open problems

OPEN 1. Equations such as $\delta_e^2 = \delta_e$ and $\delta_{aa} \subseteq \delta_a$ are 2-cells between explanations. A presentation of the 2-category (or Kleene algebra with hypotheses) they generate, and a canonical-form theorem selecting one label string per observation, is not proved here. OPEN 2. Completeness of WMRecon’s stack algorithm relative to Theorem 3. OPEN 3. We did not run the original DynamoRIO prototype and make no claim about its empirical results. A WM is an interpreter without a specification; relating reconstruction to the specializer semantics of [2] is left as a remark.

References

- [1] Robert Abela & Mark Vella (2021): Casting exploit analysis as a Weird Machine reconstruction problem. arXiv:2109.13100 [cs.CR]; arXiv metadata gives DOI 10.2053/WorldCIS.2017.0012 (WorldCIS 2017).
- [2] Third Projection Collective (2026): Computing the Futamura Diamond. <https://higher-categorical-semantics.pages.dev/research/computing-the-futamura-diamond-textual-agreement-of-generated-compilers-and-a-self-applica-e19d0b1>
- [3] Third Projection Collective (2026): The Homotopy Type of E-Graph Explanations: Measuring Proof Multiplicity in Equality Saturation. <https://higher-categorical-semantics.pages.dev/research/the-homotopy-type-of-e-graph-explanations-measuring-proof-multiplicity-in-equality-saturat-5e8806e>
- [4] Thomas Dullien (2020): Weird Machines, Exploitability, and Provable Unexploitability. IEEE Transactions on Emerging Topics in Computing 8(2), doi:10.1109/TETC.2017.2785299.
- [5] Dexter Kozen (1994): A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events. Information and Computation 110(2), doi:10.1006/inco.1994.1037.