

Specialization as Higher Rewriting: Futamura Coherence, Higher E-Graphs, and Factorization for Staging

Anonymous

Abstract. We connect Futamura projections as coherence data with higher e-graphs and cartesian comprehensive factorization. Specialization traces become proof-space paths, extensional collapse becomes 0-truncation, Conjecture A becomes a filler for an explicit diamond, and staging becomes factorization. Stabilization and trust conjectures are restated at truncation levels.

Contribution type: synthesis note. Status labels follow [A]. Definition = a stipulated structure. Observation = an immediate consequence of known facts. Proposition = a claim proved here, often by a short reduction to a result in [A] or [B]. Conjecture = an unproved claim. Provenance tags: [A] or [B] mark material taken from those notes. [new] marks synthesis introduced here. No major new theorem is claimed.

1. Question and conventions

[A] asks what coherence data the Futamura projections carry once program-equivalence witnesses are intensional. [B] asks what a rewriting system’s proof space contains beyond its congruence classes. We argue that these are one question. A specializer is a rewriting engine, and the (PE) equation is a rewrite rule. The projections are derivations, and coherence between projections is coherence between derivations in the sense of [B, §5].

Conventions follow [B]. An ∞ -category is an $(\infty, 1)$ -category, $\mathcal{S}$ is spaces, $|A|$ is groupoid completion, and $\tau_{\leq n}$ is truncation. “Cartesian” means finite-product-preserving. A functor $u : A \rightarrow B$ is initial if $|A/b| \simeq *$ for all b , and n -initial if $\tau_{\leq n}|A/b| \simeq *$ [B, Def. 4.2]. Notation from [A]: $\llbracket - \rrbracket_L$ is the semantics of L , and mix is a specializer satisfying

$$\llbracket \llbracket \text{mix} \rrbracket(p, s) \rrbracket(d) = \llbracket p \rrbracket(s, d). \quad (\text{PE})$$

2. A shared semantic setting

Definition 2.1 [new; combines A Def. 2 and B §5]. Fix a small cartesian ∞ -category T presenting the program theory. Objects are contexts and data sorts, and morphisms are program texts up to chosen identifications. A *trace system* is a set R of directed rewrite edges between parallel terms in $\text{Map}_T(\Gamma, A)$. It includes every instance of (PE) read as an edge $\llbracket \text{mix} \rrbracket(p, s) \rightsquigarrow p(s, -)$, together with a set K of specified 2-cells between parallel derivations. The *witness space* $W_{\Gamma, A}$ is the realization of the term-and-rewrite graph with the cells of K attached, as in [B, §5].

Remark 2.2. [A, Def. 2] takes 2-cells to be traces, composed by concatenation and whiskering. In Definition 2.1 a trace is a path in W , and a 2-cell of K is a path between paths. This is one dimension higher than [A]’s 2-cells. It matches [B]’s rule that a comparison of derivations is one dimension above a rewrite. With this indexing, [A]’s strict-vs-bicategory question (milestone M2) asks whether the associativity comparisons for trace concatenation lie in K or must be added.

Definition 2.3 [new]. The *observational level* of a semantics of traces is the least n such that only $\tau_{\leq n}W$ is visible. Extensional semantics is level 0: only connected components, that is congruence classes, are visible. Proof-irrelevant 2-cells in [A]’s sense are level 0 in this indexing.

3. Extensional collapse is truncation

Proposition 3.1 [new; makes A Obs. 2 precise via B Prop. 5.1]. (i) At level 0, any two derivations with the same endpoints are identified. (ii) The congruence relation generated by R does not determine W . In particular, there are finite trace systems with the same congruence but different π_1 .

Proof. (i) $\pi_0 W$ is the equivalence closure of the edges [B, Prop. 5.1]. Two parallel derivations are paths with common endpoints, so they lie in one component, and at level 0 that is all that can be seen. (ii) This is [B, Prop. 5.1] verbatim. A diamond realizes to a circle, filling it gives a disk, and both have one component. $\square$

Observation 3.2. So [A]’s Observation 2, “all routes agree trivially extensionally”, is the case $n = 0$ of a general phenomenon. Every Futamura coherence claim with content lives at level ≥ 1 .

4. The Futamura diamond and Conjecture A

Definition 4.1 (typing and rules) [new; instantiates A Def. 1, A §3 and B §5]. Work in T with one sort Tx of texts, a pairing $(-, -) : \text{Tx} \times \text{Tx} \rightarrow \text{Tx}$ given by the cartesian structure, and an application operation $\text{run} : \text{Tx} \times \text{Tx} \rightarrow \text{Tx}$ that interprets $\llbracket x \rrbracket(y)$. Literal texts are written $\ulcorner x \urcorner$, with closed terms of sort Tx . Fix closed literals mix and int . Throughout, the context is $\Gamma = (p : \text{Tx})$. The trace system R contains exactly these edges of $\text{Map}_T(\Gamma, \text{Tx})$:

- $(\text{PE}_{x,y,z})$: $\text{run}(\text{run}(\text{mix}, (x, y)), z) \rightsquigarrow \text{run}(x, (y, z))$ for terms x, y, z .
- $(\text{Ev}_{x,y})$: $\text{run}(\ulcorner x \urcorner, \ulcorner y \urcorner) \rightsquigarrow \ulcorner r \urcorner$ whenever closed evaluation of x on y terminates with output text r .
- Congruence: if $u \rightsquigarrow u'$, then $C[u] \rightsquigarrow C[u']$ for every one-hole context C built from run and pairing [B, §5].

Hypotheses. (H1) Termination. The three closed evaluations below terminate, and we name their output texts: cogen is the output of $\text{run}(\text{mix}, (\text{mix}, \text{mix}))$, comp_2 is the output of $\text{run}(\text{mix}, (\text{mix}, \text{int}))$, and comp_3 is the output of $\text{run}(\ulcorner \text{cogen} \urcorner, \text{int})$. These are the residual texts of F3, F2, and of running F3’s output on int [A, §3]. (H2) Syntactic agreement. $\text{comp}_3 = \text{comp}_2$ as texts.

Construction 4.2 (the diamond) [new]. Define four terms in context Γ , all of sort Tx :

$$\begin{aligned} s &= \text{run}(\text{run}(\text{run}(\text{mix}, (\text{mix}, \text{mix})), \text{int}), p), \\ a &= \text{run}(\text{run}(\text{mix}, (\text{mix}, \text{int})), p), \\ b &= \text{run}(\text{run}(\ulcorner \text{cogen} \urcorner, \text{int}), p), \\ c &= \text{run}(\ulcorner \text{comp}_2 \urcorner, p). \end{aligned}$$

The four edges are as follows.

1. $e_1 : s \rightsquigarrow a$ is $\text{PE}_{\text{mix}, \text{mix}, \text{int}}$ under the context $C_1 = \text{run}(-, p)$. This step is F3’s specification at int , i.e. the witness ϕ_3 of [A, Obs. 1].
2. $e_2 : a \rightsquigarrow c$ is $\text{Ev}_{\text{mix}, (\text{mix}, \text{int})}$ under C_1 . This step computes F2.
3. $e_3 : s \rightsquigarrow b$ is $\text{Ev}_{\text{mix}, (\text{mix}, \text{mix})}$ under $C_3 = \text{run}(\text{run}(-, \text{int}), p)$. This step computes F3.
4. $e_4 : b \rightsquigarrow c$ is $\text{Ev}_{\text{cogen}, \text{int}}$ under C_1 . Its target is $\text{run}(\ulcorner \text{comp}_3 \urcorner, p)$, which equals c by (H2). This step runs the generated compiler generator.

Lemma 4.3 [new]. Under (H1) and (H2), $e_1, \dots, e_4$ are well-typed edges of R . The paths $e_2 \cdot e_1$ (“specify, then compute F2”) and $e_4 \cdot e_3$ (“compute F3, then run it”) are parallel derivations $s \rightsquigarrow c$.

Proof. Typing: run and pairing are morphisms of T , and p is the only variable, so every displayed term lies in $\text{Map}_T(\Gamma, \text{Tx})$. For e_1 , substitute $x = y = \text{mix}$ and $z = \text{int}$ in PE . The left side is C_1 applied to $\text{run}(\text{run}(\text{mix}, (\text{mix}, \text{mix})), \text{int})$, i.e. s , and the right side is C_1 applied to $\text{run}(\text{mix}, (\text{mix}, \text{int}))$, i.e. a . For e_2, e_3 and e_4 , the redexes are closed applications of literals, which have the form required by Ev . They terminate by (H1), and their outputs are comp_2 , cogen and comp_3 . Placing the redexes in the stated contexts yields the stated sources and targets, with e_4 landing at c by (H2). Both composites start at s and end at c . $\square$

Remark 4.4. The pair (e_1, e_3) is a critical pair. The subterm $\text{run}(\text{mix}, (\text{mix}, \text{mix}))$ is the inner redex of e_3 and also the head of the PE -redex of e_1 . The diamond records a joining of this pair. (H2) is a genuine hypothesis:

without it, e_4 ends at a different term, and joining requires the renaming witnesses of §5. In the extensional setting, the semantic equality $\llbracket \text{comp}_3 \rrbracket = \llbracket \text{comp}_2 \rrbracket$ follows from (PE) alone. Syntactic agreement does not.

Proposition 4.5 [new; uses B Prop. 5.1]. Let $W_\diamond$ be the subcomplex of W spanned by s, a, b, c and $e_1, \dots, e_4$, together with whatever cells of K have boundary in this graph. (i) If K contains no such cell, then $W_\diamond \simeq S^1$. The two derivations of Lemma 4.3 are then not homotopic, so they are distinct witnesses at level 1, and $\pi_1(W_\diamond) \cong \mathbb{Z}$. (ii) If K contains a 2-cell θ with boundary $(e_4 \cdot e_3)^{-1} \cdot (e_2 \cdot e_1)$, then $W_\diamond$ is contractible. (iii) Restricted to $W_\diamond$, [A, Conj. A] holds if and only if such a θ exists.

Proof. (i) The graph is a 4-cycle, whose realization is a circle. The two paths differ by the generating loop, which is nontrivial in π_1 . This is the diamond of [B, Prop. 5.1]. (ii) Attaching a disk along the generating loop of a circle gives a disk. (iii) [A, Conj. A] asserts a higher cell relating the route through F3-then-F2 to the direct one. In Definition 2.1 indexing, a cell between two parallel paths in $W_\diamond$ is exactly a 2-cell bounded by their difference loop. $\square$

Remark 4.6. Proposition 4.5 does not prove Conjecture A. It isolates the finite certificate θ that milestone M2 of [A] must export, along with θ 's coherence once further diamonds are adjoined, such as associativity of PE-pastings.

Conjecture 4.7 [new; refines A Conj. A]. Consider a self-applicable offline specializer with deterministic binding-time analysis, satisfying (H1) and (H2) for all int in a decidable class. The diamonds generated by PE/Ev critical pairs then admit fillers that make each component of W 1-connected, though not necessarily contractible.

5. Stabilization up to renaming

[A, Conj. B] asserts $\llbracket \text{cogen} \rrbracket(\text{mix}) \simeq \text{cogen}$ “up to renaming”. Let G be the discrete group of renamings acting on texts.

Proposition 5.1 [new; uses B Prop. 7.1]. Let X be the space of residual texts, with G acting and with the homotopy quotient X_{hG} . For two texts x, y in one G -orbit with trivial stabilizers, the space of identifications between them is

$$\{x\} \times_{X_{hG}} \{y\} \simeq \{g \in G : gx = y\} \simeq *.$$

In general it is the set of renamings carrying x to y . After passing to the set-theoretic quotient X/G , it is always $*$.

Proof. The fibre of $X \rightarrow X_{hG}$ is G -torsorial. The pullback of two points of X_{hG} is the space of paths between them, which is the set of group elements carrying x to y . For a single free orbit this reduces to [B, Prop. 7.1], $* \times_{BG} * \simeq G$, which in π_0 becomes $*$. $\square$

Consequence. “Fixed point up to renaming” is ambiguous. In X/G it is a property. In X_{hG} it is structure: a chosen renaming g with $g \cdot \llbracket \text{cogen} \rrbracket(\text{mix}) = \text{cogen}$. Witnesses for iterated self-application then compose to a 1-cocycle. By [B, §7], one cannot quotient first and add the higher structure afterwards.

Conjecture 5.2 [new; refines A Conj. B]. For a self-applicable specializer whose output is canonical up to renaming, cogen is a homotopy fixed point in X_{hG} : the cocycle of renaming witnesses is a coboundary. Glück’s fourth-projection analysis [3] is the operational test.

6. Staging as cartesian comprehensive factorization

Definition 6.1 [new]. A *staging functor* is a product-preserving functor $F : C \rightarrow D$ of small cartesian ∞ -categories. Here C is the source theory, with contexts of the form $S \times \Delta$ (static times dynamic), and D is a residual theory. Specialization by a static value is induced by precomposition with the point $1 \rightarrow S$.

Proposition 6.2 [B, §3 applied]. Every staging functor factors as $F \simeq p_F \circ \eta_F$ in Cart_∞ . Here η_F is initial and $p_F : \int_D Y_F \rightarrow D$ is a left fibration with fibres $Y_F(d) = |C_{/d}|$, and both factors are cartesian.

Proof. This is [B, §3]. Theorem 2.3 there makes Y_F cartesian, and Lemma 2.1 transfers products. $\square$

Interpretation [new]. η_F is the binding-time-invariant part: by [B, Thm. 4.1], restricting along it changes no solution object. Every proof of semantic equality built from this part is sound. The fibre $Y_F(d)$ is the space of ways a residual object d is reached from source data. This is a residual program's space of *specialization witnesses*. At level 0, $\pi_0 Y_F(d)$ recovers the classical discrete-opfibration factor [B, §3; 5].

Proposition 6.3 [new; uses B Cor. 7.3]. If C and D have finite limits and F preserves them, both factors preserve finite limits. In particular, the equalizers used to state guards in [A, Def. 5] are preserved through staging.

Proof. This is [B, Cor. 7.3]. A guard sub-object is an equalizer, which is a finite limit. $\square$

7. Decomposition, Kan extension, and Conjecture D

[A, Conj. D] claims that specialize-then-decompose and decompose-then-specialize differ by a cost-bearing 2-cell.

Proposition 7.1 [new]. Let a program model be a product-preserving $X : C \rightarrow \mathcal{X}$ into an ∞ -topos, and let decomposition be a finite product splitting of the context. For any staging functor F , $\text{Lan}_F X$ preserves finite products. So the two orders agree on solution objects up to canonical equivalence, and the 2-cell of Conjecture D is invertible at every observational level that sees only semantics.

Proof. [B, Thm. 2.3 and Cor. 2.4] give product preservation of $\text{Lan}_F X$ for arbitrary F . Hence the image of a product splitting is a product splitting. The comparison maps between the two orders are canonical maps between products of left Kan extensions, so they are equivalences. $\square$

Consequence. The non-commutation in Conjecture D cannot be semantic. It lives in an enrichment outside the ∞ -categorical structure: the Lawvere cost enrichment of [A, Def. 8; 2]. Conjecture D is therefore a statement about a cost function on the fibres $Y_F(d)$. Different witnesses of the same residual object have different costs.

Conjecture 7.2 [new]. There is a lax cost functional $c : \int_D Y_F \rightarrow ([0, \infty], \geq, +)$ under which an optimal scheduler searches over the witness fibre rather than over orders. Conjecture D then holds exactly when c is non-constant on some fibre component.

8. Trust, descent, and the stack of verified results

[A, Conj. C] proposes a sheaf $\mathcal{V}$ of verifiable results over a network site. [B, §6] proves that local witnesses descend only with coherent overlap data (eq. (4) of [B]).

Proposition 8.1 [new]. If $\mathcal{V}$ takes values in sets (0-truncated), then gluing data over a Čech nerve reduce to equalities on double overlaps. If $\mathcal{V}$ takes values in groupoids of results-with-witnesses (for example, results modulo the renaming action of §5), then descent requires cocycle data on triple overlaps, and gluing can fail even when all double overlaps agree.

Proof. For 0-truncated Z , the limit (4) of [B] over Δ truncates to the equalizer of the first two stages. For 1-truncated Z it involves the stages up to [2]. A G -valued 1-cocycle that is not a coboundary is an example, as for nontrivial torsors. $\square$

Conjecture 8.2 [new; refines A Conj. C]. In the trust model of [A] with f Byzantine faults, the right object is a stack $\mathcal{V}_1$. Minimal inconsistent coalitions are then detected either by a failure of double-overlap agreement (0-level, as in [A]) or by a non-trivial cocycle (1-level), and the second kind is invisible to set-valued checks. Compare [4].

9. Concept correspondence

[A]	[B]	This note
(PE) equation	rewrite rule / edge	generator of R
trace 2-cell	path in Map_T	1-cell of W
coherence 3-cell (Conj. A)	comparison of derivations	2-cell of K
extensional witnesses (Obs. 2)	congruence $\pi_0 G $	level 0
strict vs. bicategory (M2)	polygraphic presentation	presented associators
up to renaming (Conj. B)	$* \times_{BG} * \simeq G$	homotopy quotient X_{hG}
staging / binding times	comprehensive factorization	initial + left fibration
guards (Def. 5)	finite-limit refinement	Prop. 6.3
decomposition order (Conj. D)	Lan preserves products	Prop. 7.1, cost enrichment
trust sheaf (Conj. C)	descent in ∞ -topos	stack, Prop. 8.1

10. Expected failure points and open problems

- Real specializers are partial, so (PE) holds only on terminating inputs [A, §7]. Definition 2.1 needs a partial or domain-enriched T .
- Proposition 7.1 assumes product-preserving models.
- No saturation algorithm or complexity bound is claimed for computing W , as in [B, §8].
- Open problems: (1) Export F1–F3 traces from a toy `mix0` and check the filler θ of Proposition 4.5 on examples. (2) Decide Conjecture 5.2 for a concrete specializer. (3) Give a presentation calculus for K (polygraphs [B, ref. 8]). (4) Construct the cost functional of Conjecture 7.2.

Authorship and automation. Submitted anonymously at the responsible researcher’s direction. An AI assistant assisted with literature retrieval, formulation, proofs, and source preparation. No independent referee review or proof-assistant certification is claimed. Publication and review classification are editorial decisions.

References

- [A] Third Projection Collective. Futamura Projections as Coherence: A Higher-Categorical Semantics of Specialization and Untrusted Distribution. <https://higher-categorical-semantics.pages.dev/research/futamura-projections-as-coherence-a-higher-categorical-semantics-of-specialization-and-unt-cd6bc55d>
- [B] Anonymous. Higher E-Graphs: Cartesian Factorization, Proof Spaces, and Topos Semantics. <https://higher-categorical-semantics.pages.dev/research/higher-e-graphs-cartesian-factorization-proof-spaces-and-topos-semantics-ba2174e4>
- [1] Y. Futamura. Partial evaluation of computation process—an approach to a compiler-compiler. Reprinted in *Higher-Order and Symbolic Computation* 12(4):381–391, 1999. doi:10.1023/A:1010095604496
- [2] F. W. Lawvere. Metric spaces, generalized logic, and closed categories. *Rend. Sem. Mat. Fis. Milano* 43:135–166, 1973. doi:10.1007/BF02924844
- [3] R. Glück. Is there a fourth Futamura projection? *PEPM 2009*. doi:10.1145/1480945.1480954
- [4] S. Abramsky, A. Brandenburger. The sheaf-theoretic structure of non-locality and contextuality. *New J. Phys.* 13:113036, 2011. doi:10.1088/1367-2630/13/11/113036
- [5] R. Street, R. F. C. Walters. The comprehensive factorization of a functor. *Bull. AMS* 79(5), 1973.
- [6] J. Lurie. *Higher Topos Theory*. Princeton University Press, 2009.
- [7] Y. Guiraud, P. Malbos. Polygraphs of finite derivation type. *Math. Struct. Comput. Sci.*, 2018.
- [8] N. D. Jones, C. K. Gomard, P. Sestoft. *Partial Evaluation and Automatic Program Generation*. Prentice Hall, 1993.