Games, Experiments, and Residual Runners

Histories, counting, and the two geometries of logical observation

Research companion to *Execution and Observation*

22 October 2026 — reviewed temporal edition

Abstract

Finite-model games have two distinct categorical presentations: game comonads unfold relational structures into bounded histories, whereas Sweedler comonads classify stateful implementations of interaction theories. Their most immediate bridge is not an isomorphism of comonads but the *behaviour map*. Pointed comodels of the free theory on Ay^B have Mealy behaviours $A^+ \to B$; after truncation at k these are exactly the underlying functions of coKleisli maps for the history comonad L_k^+, and Jakl's relational lifting selects the legal one-sided strategies. We then use a polynomial finite-list residual to execute fixed finite patterns: folding the list of successful runs computes $|\operatorname{Hom}(P, A)|$. A separate sum–product recurrence exploits an $\mathbb{E}_k$ forest cover to achieve summation nesting depth at most k; a naive DFS monad term does *not* preserve that bound. These results join Spivak's pattern–matter interaction to the finite-test and execution geometries of behaviour. A guarded Mealy construction organizes bounded responses as the stages of a terminal temporal comodel; its budgeted-word slice contains Jakl's legal one-sided strategies as a subfunctor and indeed a subcoalgebra. Finite adaptive experiments yield compositional cylinder guards and a bounded history–test pairing. Finally, the relative Plotkin–Power adjunction gives an algebraic semantics for residual observations. Established encodings and counting identities are kept distinct from the proposed uniform dependent theory, canonical game arena, and composition theorem.

1 The problem: what separates two finite structures?

A finite-model game distinguishes structures by an interactive experiment. Spoiler selects challenges, possibly adapting later challenges to previous responses; Duplicator replies subject to relational constraints. Two questions must be separated: *which challenge histories are possible?* and *which response behaviours can sustain those challenges?*

Jakl's game comonads and arboreal categories develop the first question by placing histories directly over relational structures [2, 3]. Sweedler runners develop the second by giving a stateful implementation of an algebraic theory with a residual effect [7, 8, 9]. Spivak's free polynomial monad provides the missing syntax of finite adaptive experiments, and Libkind–Spivak give an interaction in which such patterns run on continuing matter [5].

A previous version began with a history-state implementation of the Ehrenfeucht–Fraïssé game. That is a correct *encoding*, but it is close to tautological: histories as states and legal response policies are already Duplicator strategies. The task here is more exacting. We first compare the *observable behaviour* of such runners with the underlying maps in Jakl's coKleisli category. We then find a different residue for which running an experiment computes something nontrivial: homomorphism counts. This follows the programme of our preceding *Execution and Observation* interlude, where finite tests and operational continuation carry different but compatible geometries.

2 Polynomial questions, response machines, and finite patterns

For a polynomial interface

$$p = \sum_{i \in I} \mathsf{y}^{D_i},$$

I is the set of questions or challenges, and D_i the set of possible answers to question i. The free polynomial monad $\mathfrak{m}_p$ has well-founded trees as operations (finite-depth when p is finitary); at a node labelled i, one has a continuation subtree for each $d \in D_i$. Thus a tree is an adaptive experiment, not necessarily a fixed sequence of challenges.

The Dirichlet dual has the simple description

$$[p, \mathsf{y}] \cong \Big(\prod_{i \in I} D_i\Big)\mathsf{y}^I. \tag{1}$$

A coalgebra $S \to [p, \mathsf{y}](S)$ is a continuing response machine:

$$r : S \to \prod_{i \in I} D_i, \qquad \delta : S \times I \to S.$$

At each state the machine selects an answer for every possible next question and updates its state after the chosen question. A pointed coalgebra is therefore a deterministic strategy that remembers its history through S.

With a residual monad R, the corresponding polynomial internal hom is

$$[p, R](S) \cong \prod_{i \in I} R(D_i \times S). \tag{2}$$

A coalgebra $S \to [p, R](S)$ is the generator-level data of a runner from the free theory $\mathfrak{m}_p$ to the state transformer of R:

$$\rho : \mathfrak{m}_p \longrightarrow \mathrm{St}_S(R), \qquad \mathrm{St}_S(R)(X) = \big(R(X \times S)\big)^S. \tag{3}$$

Freeness makes the extension from generators unique; the monad laws of R make experiment substitution agree with successive execution. Under the polynomial free/cofree and Sweedler representing hypotheses, this also expresses the comparison

$$\boxed{\{\mathfrak{m}_p, R\} \;\cong\; \mathfrak{c}_{[p,R]}.} \tag{4}$$

Indeed, for any polynomial comonad D, maps $D \to \{\mathfrak{m}_p, R\}$ correspond to interactions of the free monad with D, hence to maps $p \to [D, R]$, hence by the Dirichlet closure to maps $D \to [p, R]$; the cofree comonad represents the latter. This reasoning is stated in the polynomial duoidal setting; it is not an unqualified assertion about arbitrary accessible endofunctors.

Libkind and Spivak establish a natural module interaction

$$\Xi_{p,q} : \mathfrak{m}_p \otimes \mathfrak{c}_q \longrightarrow \mathfrak{m}_{p \otimes q}, \tag{5}$$

where $\otimes$ is the *Dirichlet* tensor. Their evaluation with the residual dual yields

$$\boxed{\mathfrak{m}_p \otimes \mathfrak{c}_{[p,R]} \longrightarrow \mathfrak{m}_R \longrightarrow R.} \tag{6}$$

Compare the residual interaction discussed in [5]. It gives an explicit pattern–matter account of executing a finite challenge tree against a continuing residual response behaviour. It is distinct from the Plotkin–Power tensor of a comodel with an algebra, discussed in Section 9.

Remark 2.1 (Two interactions, and the Gendlin map). The earlier *Execution and Observation* interlude highlighted Spivak's Gendlin interaction

$$\mathfrak{m}_p \longrightarrow [\mathfrak{c}_p, \mathfrak{m}_{p+1}].$$

It matches a finite p-experiment against a *p-behaviour*, with an additional nullary failure possibility. Formula (6) instead matches a p-experiment against a behaviour of the *dual answering interface* $[p, R]$ and returns an R-effect. These are related polynomial constructions, but not the same map. Moreover the free polynomial monad $\mathfrak{m}_p$ should not be silently identified with the getter/update Moore monad of the earlier tutorial, even where their associated cofree implementations coincide.

3 The bounded-history comonad

Let $k \geq 1$. Define the truncated nonempty-list functor

$$L_k^+ A = \coprod_{1 \leq j \leq k} A^j.$$

Its counit returns the last element. Its comultiplication sends a history $[a_1, \ldots, a_j]$ to the history of its nonempty prefixes

$$[[a_1], [a_1, a_2], \ldots, [a_1, \ldots, a_j]].$$

The prefix equations immediately give a comonad. Given a function $f : L_k^+ A \to B$, its coKleisli extension $f^* : L_k^+ A \to L_k^+ B$ replaces each prefix by its response:

$$f^*([a_1, \ldots, a_j]) = \big[f([a_1]), f([a_1, a_2]), \ldots, f([a_1, \ldots, a_j])\big]. \tag{7}$$

This is the basic causal-dataflow interpretation of the history comonad (compare [16]).

For a relational signature σ, Jakl's $\mathbb{E}_k$ lifts this underlying comonad from **Set** to $\mathrm{Str}(\sigma)$. Its relation on histories is constrained by prefix comparability and the relations of their last elements. A coKleisli map

$$\mathbb{E}_k A \longrightarrow B$$

is consequently a history function satisfying a relational homomorphism condition, not an arbitrary Set-map.

3.1 The response behaviour of a Mealy comodel

Fix sets A, B and put $p_{A,B} = A\mathsf{y}^B$. Its dual is $[p_{A,B}, \mathsf{y}] = B^A \mathsf{y}^A$. Thus a comodel for $\mathfrak{m}_{p_{A,B}}$ is the Mealy machine

$$o : S \times A \to B, \qquad \delta : S \times A \to S. \tag{8}$$

At an initial state s_0, the response to a nonempty history $u = a_1 \cdots a_j$ is the output of the last transition, after the prefix $a_1 \cdots a_{j-1}$ has been executed. Denote it by $\mathrm{beh}_{s_0}(u)$.

Theorem 3.1 (Bounded Mealy–history comparison)**.** *For any sets A, B and $k \geq 1$, bounded observable behaviours of pointed $[A\mathsf{y}^B, \mathsf{y}]$-coalgebras are naturally in bijection with functions $L_k^+ A \to B$:*

$$\boxed{\{\textit{pointed Mealy machines}\}/\sim_{\leq k} \;\cong\; \mathbf{Set}(L_k^+ A, B),} \tag{9}$$

where two machines are equivalent when their outputs agree on every nonempty challenge history of length at most k. Under this bijection, the σ-relationally legal behaviours are precisely the underlying functions of coKleisli morphisms $\mathbb{E}_k A \to B$.

Proof. Every pointed Mealy machine gives a function by running the input history and taking its last response. Conversely, given $f : L_k^+ A \to B$, take the finite-history state space $S = A^{\leq k}$, with initial state []. At a state u of length less than k, question a returns $f(ua)$ and updates the state to ua. At states of length k assign arbitrary responses and transitions when B is nonempty; these cannot affect bounded observations. If $B = \varnothing$ and $A \neq \varnothing$, neither a Mealy machine with inhabited state nor a map $L_k^+ A \to B$ exists; the empty cases can be treated separately. Hence the observation map is surjective, and its kernel is exactly $\sim_{\leq k}$. For relational structures, by definition $\mathbb{E}_k A$ has the underlying set $L_k^+|A|$, and a function from it to B is a homomorphism exactly when it preserves the lifted relations. This is the stated legality restriction. In particular, legality depends only on the bounded *behaviour* $L_k^+ A \to B$, not on a choice of realizing state machine; it descends along $\sim_{\leq k}$. □

Remark 3.2 (Exactly what is compared). The theorem is a comparison of *behaviours*, not an equivalence of the two comonads. Jakl's $\mathbb{E}_k$ acts on the structure of challenges and preserves its relational information. The Sweedler comonad $\{\mathfrak{m}_{Ay^B}, \mathsf{y}\}$ acts on response states. The bridge is evaluation of the latter on bounded challenge histories, followed by Jakl's relational constraint. The history realization in the proof is not claimed to be minimal.

Remark 3.3 (Forth-only versus back-and-forth). A homomorphism $\mathbb{E}_k A \to B$ describes a one-sided, relation-preserving response behaviour. The full EF game involves moves in both directions, equality consistency, and reflection of atomic relations. It is not characterized by arbitrary individual coKleisli maps. We recover it using a two-sided interface and a certified safety condition in Section 5.

4 Finite game horizons and the temporal geometry of strategies

The comparison in Theorem 3.1 was made one horizon at a time. The bounded-comodel construction of *Sweedler Runners for Temporal Systems, I* [1] suggests a more coherent answer: the round bounds are stages of a single guarded response machine. This section proves that answer for the free Mealy theory, then relates the resulting budgeted paths to Jakl's legal coKleisli strategies. No graded *residual* Sweedler hom is assumed here.

4.1 A guarded Mealy comodel and its terminal horizons

Let $\mathcal{E} = \mathbf{Set}^{\mathbb{N}^{\mathrm{op}}}$, whose objects are inverse sequences $S_0 \leftarrow S_1 \leftarrow \cdots$ with restriction maps $r_{n+1,n}$. Write ΔB for the constant object and $\triangleright$ for the later functor,

$$(\triangleright S)_0 = 1, \qquad (\triangleright S)_{n+1} = S_n.$$

The correct guarded version of the response interface is

$$\boxed{G_{A,B}(S) = \big(\triangleright(\Delta B \times S)\big)^{\Delta A}.} \tag{10}$$

The later modality applies to *both* the response and the next state. Indeed $G_{A,B}(S)_0 = 1$, whereas $G_{A,B}(S)_{n+1} = (B \times S_n)^A$. In particular, a stage-zero state is not required to answer a challenge that its budget cannot execute. A coalgebra $S \to G_{A,B}(S)$ has for each $n \geq 0$ and $a \in A$ response and update maps

$$\beta_{a,n} : S_{n+1} \to B, \qquad \delta_{a,n} : S_{n+1} \to S_n,$$

whose compatibility with restrictions, for all $n \geq 0$, is

$$\beta_{a,n+1} = \beta_{a,n} r_{n+2,n+1}, \qquad r_{n+1,n}\delta_{a,n+1} = \delta_{a,n} r_{n+2,n+1}. \tag{11}$$

We call this a *bounded Mealy comodel*. It is the response-side counterpart of the graded Moore comodel constructed in the temporal note, not an assertion that the two operational monads are identical.

Theorem 4.1 (Terminal bounded response behaviours)**.** *The final $G_{A,B}$-coalgebra is the inverse sequence*

$$\boxed{\Omega_0 = 1, \qquad \Omega_n = B^{L_n^+ A} \quad (n \geq 1), \qquad L_n^+ A = \coprod_{1 \leq j \leq n} A^j.} \tag{12}$$

Restriction forgets response coordinates on words of length $n+1$. The response to a is $f(a)$ and the residual update is

$$\partial_{a,n}(f)(v) = f(av) \quad (f \in \Omega_{n+1},\ 1 \leq |v| \leq n).$$

There is a natural isomorphism $\Omega_{n+1} \cong (B \times \Omega_n)^A$. For every bounded Mealy comodel S, the unique coalgebra morphism $\mathrm{beh} : S \to \Omega$ *sends $s \in S_n$ to the responses generated by executing every nonempty challenge word of length at most n.*

Proof. A response function on words of lengths $1, \ldots, n+1$ is uniquely specified by the response to each first challenge a, and by its responses to the nonempty words av of lengths at least two. These are exactly a member of $(B \times \Omega_n)^A$. The displayed structure defines a $G_{A,B}$-coalgebra. Given any other coalgebra, construct beh_n recursively using $\beta_{a,n-1}$ for words of length one and $\delta_{a,n-1}$ followed by beh_{n-1} for longer words. Equations (11) make this compatible with restriction; the same recursion proves uniqueness. No finiteness of A or B is needed. □

Example 4.2 (What a second round adds)**.** For $A = \{a\}$ and $B = \{0, 1\}$, the first stages are $\Omega_0 = 1$, $\Omega_1 = 2$, $\Omega_2 = 2^2$, and $\Omega_3 = 2^3$. A point of Ω_2 records the answers to a and aa. The restriction $\Omega_2 \to \Omega_1$ forgets the second answer, whereas the execution arrow labelled a returns a one-round future whose answer is the original answer to aa. These are two different ways of going from budget two to budget one.

4.2 Budgeted words and the slice of strategies

Define a category $\mathcal{J}_A$ with objects $n \in \mathbb{N}$ and

$$\mathcal{J}_A(n, m) = \begin{cases} A^{\leq n-m}, & n \geq m, \\ \varnothing, & n < m. \end{cases} \tag{13}$$

An arrow $w : n \to m$ first executes the challenge word w and then discards the unused allowance $n - m - |w|$; composition is word concatenation in execution order. Thus distinct arrows labelled ε and a can have the same budget endpoints. For $f \in \Omega_n$ define

$$(\sigma_w f)(v) = f(wv) \quad (1 \leq |v| \leq m). \tag{14}$$

When $m = 0$ this is the unique point of Ω_0. The length inequality in (13) makes the formula meaningful.

Theorem 4.3 (Budgeted-word slice and legal subfunctor)**.** *There is a natural equivalence*

$$\boxed{\mathrm{Coalg}_{\mathcal{E}}(G_{A,B}) \simeq [\mathcal{J}_A, \mathbf{Set}]/\Omega \simeq [\textstyle\int_{\mathcal{J}_A} \Omega, \mathbf{Set}].} \tag{15}$$

Here $\Omega : \mathcal{J}_A \to \mathbf{Set}$ acts as in (14). If A, B are relational structures over one signature, set

$$\Lambda_0(A, B) = 1, \qquad \Lambda_n(A, B) = \mathrm{Hom}_{\mathrm{Str}(\sigma)}(\mathbb{E}_n A, B) \quad (n \geq 1).$$

Then the forgetful embeddings $\Lambda_n(A,B) \hookrightarrow \Omega_n$ *assemble into a subfunctor*

$$\boxed{\Lambda(A,B) \hookrightarrow \Omega \quad \textit{in } [\mathcal{J}_A, \mathbf{Set}].} \tag{16}$$

Thus legal one-sided strategies at all horizons form a subobject of the terminal bounded response behaviour in the budgeted-word presentation.

Proof. A guarded Mealy coalgebra has restrictions and update maps obeying (11). Hence every arrow of $\mathcal{J}_A$ acts on S by its updates followed by restriction. The final behaviour maps constitute a natural transformation $S \to \Omega$. Conversely, given a functor $S : \mathcal{J}_A \to \mathbf{Set}$ over Ω, its empty-word arrows give restrictions, its one-letter arrows give updates, and its map to Ω gives the response $\beta_{a,n}(s) = \mathrm{beh}_{n+1}(s)(a)$. Naturality supplies (11); the two constructions are inverse. The category-of-elements equivalence proves the second part of (15).

For the legal subfunctor, a word $w : n \to m$ gives a map of relational structures

$$j_w : \mathbb{E}_m A \longrightarrow \mathbb{E}_n A, \qquad v \longmapsto wv \quad (m \geq 1).$$

Appending a fixed common prefix preserves comparability of histories; their last elements are unchanged, so all lifted relations are preserved. Hence j_w is a homomorphism. Precomposition $f \mapsto f \circ j_w$ maps $\Lambda_n(A,B)$ to $\Lambda_m(A,B)$ and is exactly the restriction of σ_w. At $m = 0$ use the unique map to 1. These maps satisfy the concatenation law and preserve identity arrows. This proves (16). □

Corollary 4.4 (The universal legal one-sided responder)**.** *For fixed relational structures* A, B*, the inclusions* $\Lambda_n(A,B) \subseteq \Omega_n$ *constitute a subcoalgebra*

$$\boxed{\Lambda(A,B) \hookrightarrow \Omega \quad \textit{in } \mathrm{Coalg}_{\mathcal{E}}(G_{A,B}).}$$

Consequently, for any bounded Mealy coalgebra S *with terminal behaviour map* $\mathrm{beh} : S \to \Omega$*, the following are equivalent: (i) every stagewise state has a relationally legal bounded behaviour; (ii)* beh *factors through* Λ *as a coalgebra morphism. More generally the pullback*

$$\boxed{S^{\mathrm{leg}} := S \times_\Omega \Lambda \hookrightarrow S}$$

is a subcoalgebra and is the largest subcoalgebra of S *all of whose states have legal one-sided behaviours. In the arrow-topos reading, its inclusion is the canonical safety predicate on response states; proof-relevant legality witnesses would require an additional witness object.*

Proof. The guarded Mealy cooperation of $f \in \Omega_{n+1}$ returns, at each question a, the pair $(f(a), \sigma_a f) \in B \times \Omega_n$. The subfunctor property in Theorem 4.3 says that $\sigma_a f \in \Lambda_n$ whenever $f \in \Lambda_{n+1}$. Thus the terminal cooperation restricts to Λ, and its inclusion is a coalgebra morphism. The factorization statement is immediate. The functor $G_{A,B}$ preserves pullbacks (stagewise, restriction/guarding preserves finite limits, and products and exponentials preserve pullbacks). Hence pulling the subcoalgebra back along beh gives a subcoalgebra of S. Any legal subcoalgebra must factor into this pullback, proving maximality. □

Remark 4.5 (The precise scope of legality). The subfunctor records the *one-sided*, relation-preserving coKleisli strategies of Jakl's lifted history comonad. In the full Ehrenfeucht–Fraïssé game, the chosen answers must also reflect relations and equality against *previously played* pairs. A residual response function by itself need not retain the previous cross-structure matching; the state or index must retain the played position. A uniform two-sided construction therefore belongs to the certified/dependent runner problem, not to the bare subfunctor statement above.

4.3 Guarded adaptive experiments

For this subsection let A, B be finite, with $B \neq \varnothing$. A *binary-labelled finite experiment* of depth at most n is a finite tree whose internal nodes carry challenges $a \in A$, with one outgoing branch for each response $b \in B$, and whose leaves are labelled 0 or 1. In Spivak's language these are elements of the corresponding free polynomial monad $\mathfrak{m}_{Ay^B}(2)$. Running an experiment against $f \in \Omega_n$ selects a unique leaf: a challenge prefix $a_1 \cdots a_j$ receives answer $f(a_1 \cdots a_j)$. Write $e_t(f) \in 2$ for that leaf's label.

For each leaf ℓ of t, suppose its path has challenge labels $a_1, \ldots, a_d$ and successive reply labels $b_1, \ldots, b_d$. The *guard* of that leaf is

$$U_{t,\ell} = \{f \in \Omega_n : \ f(a_1 \cdots a_j) = b_j \ (1 \leq j \leq d)\}. \tag{17}$$

It describes the finite experimental evidence that forces execution to take precisely that path.

Proposition 4.6 (Finite game guards and substitution)**.** *For each finite adaptive test t, the guards $U_{t,\ell}$ are cylinders partitioning Ω_n. For finite A, B, Ω_n is finite discrete, so they are automatically clopen. If a test u of depth at most $n - d$ is substituted at a leaf ℓ of depth d, then for every leaf j of u,*

$$\boxed{U_{t \circ_\ell u,(\ell,j)} = U_{t,\ell} \cap \sigma_{w_\ell}^{-1}(U_{u,j}), \quad w_\ell = a_1 \cdots a_d.} \tag{18}$$

Moreover, the family of binary-valued depth-$\leq n$ tests separates points of Ω_n.

Proof. At each challenge node every response labels exactly one branch, so each f picks exactly one leaf. Formula (17) uses finitely many coordinates and is therefore a cylinder; clopenness at finite n is immediate from discreteness. After substitution, a behaviour reaches leaf (ℓ, j) exactly when it first follows ℓ, and its residual behaviour $\sigma_{w_\ell} f$ then follows j. This proves (18), as well as the pullback-composition law for the associated guarded temporal paths.

If $f \neq g$ in Ω_n, choose a nonempty word $w = a_1 \cdots a_d$ of length $d \leq n$ where their responses differ. Take the test which asks $a_1, \ldots, a_d$ in succession, independently of the earlier replies, and labels its final leaves by 1 exactly when the last reply equals $f(w)$. Then $e_t(f) = 1$ and $e_t(g) = 0$. This proves separation. □

Remark 4.7 (Where the Stone topology first matters). At a fixed finite horizon with finite A, B, calling the guards clopen says little: Ω_n is already finite discrete. Their topological significance emerges in the limit

$$\Omega_\infty = B^{A^+} = \varprojlim_n \Omega_n.$$

The guards then become basic finite-coordinate clopen cylinders of the product topology, which is Stone for finite discrete B. For the finite dual polynomial $q = [A\mathsf{y}^B, \mathsf{y}]$, this is an instance of Spivak's topology on $\mathfrak{c}_q(1)$ generated by finite tree tests [13]. Our *Spoiler* trees come from $\mathfrak{m}_{Ay^B}$ rather than $\mathfrak{m}_q$; the equality of the induced finite-coordinate topology in this special Mealy example follows from their finite support and the separating tests above, not from an identification of the two tree syntaxes.

4.4 A bounded history–test pairing, and the Chu question

For $f \in \Omega_k$, a challenge prefix $w \in A^{\leq k}$ and a test t of depth at most $k - |w|$, define

$$\boxed{H_k(f; w, t) = e_t(\sigma_w f).} \tag{19}$$

A fixed history selects a residual strategy; the future test is run against that residual. This is the finite-budget form of the history–test pairing in Section 8. For test substitution, the guard identity (18) is the precise bounded analogue of the identity "execute, then test = extend the test." Tests separate bounded response behaviours, but their restriction to the subfunctor Λ need not expose the additional relational certificates as independent data.

The temporal Moore note proves a temporal-valued balanced tensor and both Chu transposes,

$$\mathcal{A}(B_0, [S, K]) \cong \mathcal{E}(B_0 \widehat{\otimes} S, K) \cong \mathcal{C}(S, \langle B_0, K \rangle),$$

for its *specific graded Moore* algebra and comodel categories [1]. The guarded leaf calculus above gives a candidate Mealy counterpart: finite experiments are the algebraic inputs, Ω supplies bounded responses, and $\Delta 2$ supplies Boolean outcomes. However, a graded Mealy tensor with the required substitution balancing, and its second transpose, have not been constructed here. Nor does that ordinary Boolean Chu tensor yet interpret a residual list or exception effect. These are separate extensions, not consequences of the Moore theorem by notation.

Remark 4.8 (A graded residual target). The suggested generalization, still to be axiomatized, uses cooperations of the schematic form

$$\theta^R_{m,n,X} : T_m X \times S_{m+n} \longrightarrow R(X \times S_n).$$

One must impose unit, substitution, grade weakening and horizon restriction laws, and specify how the residual effect handles continuations that terminate or branch. A List residue might enumerate several paths of different lengths; an exception residue may terminate before its allotted horizon. A fully satisfactory theory should account for discarded allowance, and need not be identical to internal Sweedler measurement in $\mathcal{E}$.

5 A back-and-forth game as certified residual execution

Let A, B now be finite relational structures. Set

$$p_{A,B} = \sum_{a \in A} \mathsf{y}^B + \sum_{b \in B} \mathsf{y}^A, \qquad T = \mathfrak{m}_{p_{A,B}+1}, \qquad R(X) = X + E, \quad E = \{\mathsf{win}, \mathsf{lose}\}. \tag{20}$$

A challenge $a \in A$ has a B-indexed choice of continuations; a challenge $b \in B$ has an A-indexed choice. The additional nullary generator done terminates a test. It is precisely here that a pure Set-comodel fails: a nullary operation would need a map $S \to \varnothing$. A residual runner can instead interpret it as $S \to R0 = E$. (Cf. [8] for operational exceptions and termination in runners.)

A position is a finite sequence of pairs $p = ((a_1, b_1), \ldots, (a_j, b_j))$. It is *legal* if the correspondence of named elements is a partial isomorphism, including equality. A state records a legal position and the remaining round budget. A certified runner is a monad morphism

$$\rho : T \to \mathrm{St}_S(R)$$

whose generator implementation either gives a legal response with an updated position, or gives lose when no legal response exists; done returns win. Only tests making at most the available number of challenges are admitted. In particular a runner is not allowed to report arbitrary early victory as a way to dodge a challenge.

Write W_n for the positions at which Duplicator can survive n more rounds. For the fixed budget k and legal positions of length at most k, put

$$\begin{aligned} W_0 &= \{p \mid p \text{ is legal and } |p| \le k\}, \\ p \in W_{n+1} &\iff [p \text{ legal and } |p| + n + 1 \le k] \\ &\quad \wedge\ \forall a \in A\ \exists b \in B : p + (a,b) \in W_n \\ &\quad \wedge\ \forall b \in B\ \exists a \in A : p + (a,b) \in W_n. \end{aligned} \tag{21}$$

(The order of the pair is always (a, b), regardless of which structure Spoiler challenged.)

Proposition 5.1 (EF as an encoding by residual runners)**.** *For finite A, B and k, Duplicator wins the k-round EF game if and only if there is a certified residual runner whose every admissible closed challenge test of depth at most k returns* win *from the empty position. Equivalently, $[] \in W_k$. If $[] \notin W_k$, there is a finite adaptive Spoiler tree of depth at most k, with terminal leaves* done*, that makes every certified response policy return* lose*.*

Proof. The first equivalence is the usual back-and-forth recursion written as generator-level runner data: from positions in W_n, choose a reply in W_{n-1} for each challenge, and use the response functions to define the runner. The freeness of T extends these operations to arbitrary tests. Conversely, a runner that wins every permitted test must respond to each challenge with a state able to survive all shorter continuations, so induction on n gives membership in W_n.

For the separating tree one must include the budget condition in the induction. Prove the claim only for positions p with $|p| = k - n$ and $p \notin W_n$ (the positions reached at the corresponding stage from the initial game). When $n > 0$ the inequality $|p| + n \le k$ is now automatic. Consequently one of the two universal clauses in (21) fails. Choose the corresponding challenge. Every legal answer leads to a position outside W_{n-1}; attach its separating tree to that branch. A challenge with no legal answer yields lose immediately. At $n = 0$, the only losing outcome arises from a failure already recorded by the residual. This recursive construction builds a single Spoiler term against every certified policy. □

Example 5.2 (One element versus two)**.** Let $A = \{*\}$ and $B = \{0, 1\}$, with no non-equality relations. Duplicator survives one round. In two rounds Spoiler chooses 0 then 1 in B; Duplicator must answer $*$ twice, violating equality reflection. The corresponding finite test returns lose for every certified policy.

Remark 5.3 (What the proposition does and does not prove). This is an *encoding theorem*, not a new proof of the EF characterization of quantifier rank. The genuinely useful result so far is Theorem 3.1: it identifies how Sweedler response behaviours compare with Jakl's lifted history maps. The present construction also isolates the residue needed to report failed or completed plays.

6 Counting homomorphisms by a list residual

For homomorphism counts, failure/success as a Boolean outcome is not enough. We want a residual that preserves *all* successful assignments, with multiplicity. A writer $\mathbb{N} \times -$ accumulates a weight along one chosen run, but does not by itself sum over alternative choices. We instead choose the polynomial finite-list monad

$$\boxed{R(X) = \mathrm{List}(X) = \coprod_{n \ge 0} X^n.} \tag{22}$$

Its monad multiplication concatenates lists. The empty list is failure, while many elements are alternative successful continuations.

Fix a finite relational pattern P and a finite target structure A, and enumerate $V(P) = \{v_1, \dots, v_m\}$. Consider the polynomial

$$p_{P,A} = \sum_{v \in V(P)} \mathsf{y}^A, \qquad T_{P,A} = \mathfrak{m}_{p_{P,A}}.$$

Let S be the set of partial homomorphism assignments $s : U \to A$ with $U \subseteq V(P)$: every relation tuple whose entries lie in U is respected. For this section assume either that the signature has no nullary relations, or that the nullary facts of P are true in A; otherwise the entire count is zero and can be handled by a scalar preliminary check.

For a generator asking for the image of v, define

$$\delta_v(s) = \begin{cases} [(s(v), s)], & v \in \mathrm{dom}(s), \\ [\,(a, s[v \mapsto a]) \mid a \in A, s[v \mapsto a] \text{ partial hom.}\,], & v \notin \mathrm{dom}(s). \end{cases} \tag{23}$$

Fix an order on A to order the list; its length is independent of that order. This gives a coalgebra $S \to [p_{P,A}, \mathrm{List}](S)$ and hence a residual runner $\rho : T_{P,A} \to \mathrm{St}_S(\mathrm{List})$.

Let $t_{P,A} \in T_{P,A}(1)$ be the *non-adaptive* sequential test: ask v_1, then v_2, and so on to v_m, with every reply branch leading to the same next question, and finally return the unique element $* \in 1$. It is important that this is a term of $T(1)$, not a closed term in $T0$. When $A \neq \varnothing$, $p_{P,A}$ has no nullary operations, so the free monad has $T_{P,A}(0) = \varnothing$: a closed term literally does not exist. The returned value at a leaf distinguishes success from the empty list. (For $A = \varnothing$ and nonempty $V(P)$, the generator arities instead become nullary.)

Theorem 6.1 (Counting by residual execution)**.** *For every finite relational P, A (with the preceding nullary convention),*

$$\boxed{\#\rho_1(t_{P,A}, \varnothing) = |\,\mathrm{Hom}_{\mathrm{Str}(\sigma)}(P, A)|.} \tag{24}$$

Here $\#\rho_1(t_{P,A}, \varnothing)$ is the length of the finite list of successful values and final states, and the initial state is the empty assignment.

Proof. After asking $v_1, \dots, v_j$, the list generated by the runner has one entry for each partial homomorphism $P|_{\{v_1,\dots,v_j\}} \to A$, with its chosen partial assignment as state; every such assignment occurs exactly once. This holds at $j = 0$. At stage $j + 1$, the runner appends once for each compatible value of v_{j+1} and discards incompatible extensions, giving the induction step. At $j = m$ the assignments are precisely full homomorphisms $P \to A$. Their number is the length of the resulting list. □

Example 6.2 (A two-edge directed path)**.** Let P have vertices u, v, w and edges $u \to v$ and $v \to w$. The runner asks for their images and returns one successful branch for every directed walk of length two in A:

$$|\,\mathrm{Hom}(P, A)| = \sum_{a,b,c \in A} [E_A(a, b)]\,[E_A(b, c)] = \sum_{b \in A} \deg^-(b) \deg^+(b).$$

The list retains individual successful assignments; its length forgets their names but records multiplicity.

6.1 Summation as a genuine handler

The free List-algebra on 1 is the free monoid of singleton words. More usefully, the commutative monoid $(\mathbb{N}, +, 0)$ is a List-algebra through

$$h : \mathrm{List}(\mathbb{N}) \to \mathbb{N}, \quad h([n_1, \ldots, n_j]) = \sum_i n_i.$$

The runner induces a $T_{P,A}$-algebra structure on $\mathbb{N}^S$:

$$(\mathsf{ask}_v(f_a)_{a\in A})(s) = \sum_{(a,s')\in\delta_v(s)} f_a(s'). \tag{25}$$

Interpreting the final return by the constant function 1 computes exactly the count in Theorem 6.1. This is a use of a *non-free* residual algebra: list concatenation is interpreted by addition rather than by a raw list of runs.

There is more algebraic information. If $v \neq w$, assigning v then w or w then v gives the same admissible pairs and the same final partial assignment. Therefore the induced $T_{P,A}$-algebra $\mathbb{N}^S$ validates the interchange equations

$$\mathsf{ask}_v\big(\mathsf{ask}_w(x_{a,b})_{b\in A}\big)_{a\in A} = \mathsf{ask}_w\big(\mathsf{ask}_v(x_{a,b})_{a\in A}\big)_{b\in A} \quad (v \neq w). \tag{26}$$

This equality holds for *all* continuations $x_{a,b} \in \mathbb{N}^S$: both sides sum the same contributions. The raw List runner generally does not satisfy literal equality of lists, since the enumerations have different orders. Thus the handler forgets precisely the order information required for interchange. We return to this distinction in Section 9.

Proposition 6.3 (A multiset residual makes interchange literal)**.** *Let* $\mathsf{Bag}(X) = \coprod_{n\geq 0} X^n/\mathfrak{S}_n$ *be the finite-multiset monad. Quotienting the lists produced by the counting runner through the monad morphism* List $\to$ Bag *gives a residual runner satisfying the interchange equation* (26) *already in its state-passing multiset semantics, for distinct vertices* v, w*.*

Proof. The two orders of questions enumerate the same pairs of compatible assignments $(v \mapsto a, w \mapsto b)$, with the same final state and the same variable $x_{a,b}$, possibly in different orders. Finite multisets forget precisely that enumeration order while retaining multiplicities, so the outcomes agree for arbitrary continuations. The corresponding lists need not be equal. □

Remark 6.4 (Analytic rather than polynomial). The multiset monad is analytic, not polynomial in general: its n-ary operations have symmetric-group quotients. It is therefore a better *literal* commutative counting residue, but taking it into the polynomial Sweedler/Dirichlet calculus requires extending the ambient setting. In particular, one cannot assume without proof that polynomial duoidal interchange is inherited by this analytic quotient.

7 Forest covers and the resource bound

The previous counting theorem does not preserve the resource that Jakl's comonad measures. A forest of height k can have arbitrarily many vertices. A DFS order still produces a sequential term of depth $|V(P)|$. We need a different syntax that expresses independent subtrees multiplicatively, rather than pretending that they are sequentially dependent.

Let F be a rooted forest on $V(P)$ such that, whenever two vertices occur together in a relation tuple of P, they are comparable in the ancestor order. Such a forest is a tree-depth cover,

and is exactly the kind of decomposition underlying $\mathbb{E}_k$-coalgebra structures (with the usual relational and equality qualifications) [4, 2]. Suppose its height is at most k.

For every vertex v, let $\mathrm{Anc}(v)$ be its strict ancestors. If $\alpha : \mathrm{Anc}(v) \to A$ is an assignment, write $\chi_v(\alpha, a) \in \{0, 1\}$ for the conjunction of all relational constraints whose *deepest distinct vertex* is v, evaluated under α together with $v \mapsto a$. Because the vertices occurring in each relation tuple form a chain, each non-nullary tuple has a unique deepest distinct vertex. Hence every relational constraint is tested exactly once.

Define values recursively from leaves towards roots:

$$\boxed{Z_v(\alpha) = \sum_{a \in A} \chi_v(\alpha, a) \prod_{w \in \mathrm{Child}(v)} Z_w(\alpha \cup \{v \mapsto a\}).} \tag{27}$$

An empty product is 1. If $r_1, \ldots, r_j$ are the roots, set $Z_F(A) = \prod_i Z_{r_i}(\varnothing)$ (also 1 for the empty forest, subject to the nullary check).

Theorem 7.1 (Resource-preserving sum–product evaluation)**.** *For a finite relational pattern P with a forest cover F as above,*

$$\boxed{Z_F(A) = |\operatorname{Hom}(P, A)|.} \tag{28}$$

Moreover the expression (27) has at most $\mathrm{height}(F) \le k$ *nested summation levels along any root-to-leaf dependency branch; independent child subtrees are combined by multiplication.*

Proof. Fix an assignment α on the strict ancestors of v. For each choice a for v, the factor $\chi_v(\alpha, a)$ checks exactly the constraints completed at v. Constraints involving vertices in different child subtrees cannot exist: such vertices are incomparable in the forest. Once the ancestor assignment and $v \mapsto a$ are fixed, the choices of homomorphism assignments in distinct child subtrees are therefore independent, and their counts multiply. Summing over all images $a \in A$ gives the number of admissible assignments to the subtree of v. Induction upwards establishes (27); multiplication over the roots gives $|\operatorname{Hom}(P, A)|$. The height claim follows immediately from the recursive syntax: every move to a child adds at most one nested sum, whereas the product combines sibling expressions without nesting them. □

Example 7.2 (A star exposes the difference)**.** Let P be the directed star with root r and leaves $v_1, \ldots, v_m$, with an edge $r \to v_i$ for every i. Its tree-depth is 2, whatever m. The naive sequential test makes $m + 1$ asks. The sum–product test has only two levels:

$$|\operatorname{Hom}(P, A)| = \sum_{a \in A} \left(\sum_{b \in A} [E_A(a, b)] \right)^m .$$

The independent choices of leaf images are multiplied, not nested one after another.

Remark 7.3 (Scope of the theorem)*.* This theorem is an explicit counting algorithm on a supplied forest cover. It is *not yet* a representation of the sum–product syntax inside the ordinary free monad $\mathfrak{m}_p$ that preserves its height. Such a lift would need a product/composition principle for independent subtests, or a distributive/graded construction. It is the next structural task, not an established corollary.

Remark 7.4 (A duoidal candidate for resource-preserving tests)*.* The free polynomial monad uses substitution $\triangleleft$ to express sequential dependence. In a forest test, the choices along an ancestor chain are sequential, but disjoint child subtrees are independent. A candidate syntax should therefore track both polynomial substitution $\triangleleft$ (whose depth measures the logical

budget) and Dirichlet tensor $\otimes$ (which combines independent branches without increasing that depth). The duoidal interchange has a comparison of the form

$$(p \otimes q) \triangleleft (r \otimes s) \longrightarrow (p \triangleleft r) \otimes (q \triangleleft s).$$

The equality of sum–product evaluations in (26) is suggestive of a semantic shadow of such interchange. This is a *candidate programme*: the recurrence in Theorem 7.1 alone does not construct a free $\triangleleft/\otimes$ game theory, nor prove that it retains forest height under all substitutions. The analytic multiset residual in Proposition 6.3 makes this question especially interesting beyond strict polynomial monads.

Dawar–Jakl–Reggio's Lovász-type results connect homomorphism indistinguishability over bounded tree-depth test structures to corresponding counting logics [10]; recent work further analyzes combined variable and depth bounds [11]. Our calculation *implements* the homomorphism-counting tests; it does not reprove their logical-equivalence theorems or infer counting logical equivalence from adaptivity alone.

8 Two geometries of game behaviour

The preceding *Execution and Observation* interlude treated Moore systems: finite experiments generated one topology on the terminal behaviour space, and operational continuation generated another. Its compatibility identity was

$$e_v \circ \sigma_u = e_{uv}.$$

We can now give a precise one-sided game analogue.

For fixed finite alphabets A, B, the terminal Mealy response space is

$$\mathcal{B} = B^{A^+}.$$

For $u \in A^*$ and nonempty $v \in A^+$, define

$$(\sigma_u b)(v) = b(uv), \qquad e_v(b) = b(v). \tag{29}$$

The empty word acts as the identity. Finite sets of observations e_v generate the product (finite-test) topology τ_{test} when B is discrete.

Theorem 8.1 (Execution–test compatibility for response behaviours)**.** *For all $u \in A^*$ and $v \in A^+$,*

$$\boxed{e_v \circ \sigma_u = e_{uv}.} \tag{30}$$

Consequently each execution shift σ_u is continuous for τ_{test}. For a pointed Mealy machine with response behaviour b_0, its reachable–observable quotient has carrier

$$\boxed{\{\sigma_u b_0 : u \in A^*\} \subseteq \mathcal{B},} \tag{31}$$

which is the least forward-invariant subset containing b_0. The restricted evaluations e_v separate the points of this minimal realization.

Proof. Both sides of (30) evaluate at b to $b(uv)$, so preimages of finite-test cylinders under shifts are finite-test cylinders. For a pointed machine, the behaviour of the state reached after u is $\sigma_u b_0$. The image of the reachable-state set in the final Mealy coalgebra is therefore exactly (31); quotienting by behavioural equivalence identifies states with the same image. Distinct response functions have different values at some nonempty history v and are separated by e_v. □

If we equip $\mathcal{B}$ with the Alexandrov topology of forward reachability, its opens are the sets closed under all shifts. Using the underlying-topological-space construction of Fairbanks–Carlson–Spivak [15], the description as forward-closed subsets is our calculation for this particular polynomial cofree comonad, rather than a general theorem about all comonads or a statement we attribute verbatim to those authors. Exactly as for Moore behaviours, the two topologies generally differ.

For a specific pointed response behaviour define its Hankel pairing

$$\boxed{H_{b_0}(u,v) = e_v(\sigma_u b_0) = b_0(uv), \qquad u \in A^*,\ v \in A^+.} \tag{32}$$

Histories index reachable residual behaviours; future experiments index columns that distinguish them. The formula directly joins the game setting to the minimal-realization theorem in *Execution and Observation*.

8.1 Residual behaviours and observable outcomes

For $R = \mathrm{List}$ or $R = (-) + E$, the final response space is instead

$$\mathcal{B}_{p,R} = \mathfrak{c}_{[p,R]}(1) \cong \{\mathfrak{m}_p, R\}(1)$$

in the polynomial setting. A finite challenge pattern and the interaction (6) produce an R-observation, whose externally visible component must be chosen. The resulting tests need not separate every final behaviour. For example, observing *only list length* identifies behaviours which produce different lists with the same multiplicity.

When the residual can terminate, a shift σ_h is defined only for executable challenge histories h, and the extended-test identity takes the typed form

$$e_t(\sigma_h b) = e_{h;t}(b), \quad \text{when continuation along } h \text{ exists.}$$

For a general branching residual, continuation after a challenge may itself be a family of residual behaviours, not a single partial shift. In that case the correct generalization is an interaction relation or a Kleisli-valued continuation, rather than an unqualified equality of functions. This is the point at which non-representable halos and joint residual observations, discussed in the preceding interlude, become relevant.

9 The relative Plotkin–Power tensor

The Dirichlet tensor $\otimes$ in (5) pairs polynomial patterns with polynomial behaviours. The Plotkin–Power tensor instead pairs a comodel with an algebra and produces a Set-valued functor [6]. Residual runners give a relative version that lands in residual algebras.

Fix any Set-monads T, R, a set S, and a runner $\rho : T \to \mathrm{St}_S(R)$. For an R-algebra $h : RX \to X$, induction along ρ equips X^S with a T-algebra:

$$(I_\rho h)(t)(s) = h(R(\mathrm{ev})(\rho_{X^S}(t,s))), \qquad \mathrm{ev} : X^S \times S \to X. \tag{33}$$

This gives a functor $I_\rho : \mathrm{Alg}(R) \to \mathrm{Alg}(T)$.

Theorem 9.1 (Relative Plotkin–Power adjunction)**.** *For any Set-monads T, R and runner ρ as above, the functor I_ρ has a left adjoint*

$$\boxed{L_\rho : \mathrm{Alg}(T) \rightleftarrows \mathrm{Alg}(R) : I_\rho.}$$

For a T-algebra $\beta : TB \to B$, it admits the presentation

$$F_R(S \times TB) \rightrightarrows F_R(S \times B) \longrightarrow L_\rho B, \tag{34}$$

with one map induced by β and the other by executing the runner and flattening the residual free algebra. On free algebras,

$$\boxed{L_\rho F_T(X) \cong F_R(S \times X).} \tag{35}$$

The adjunction unit at $F_T(X)$ is the original running map $T(X) \to (R(X \times S))^S$, up to the canonical swaps.

Proof. The monad $\mathrm{St}_S(R)$ is induced by composing the adjunctions $(- \times S) \dashv (-)^S$ on **Set** and $F_R \dashv U_R$. Hence induction can be viewed as restriction along ρ following the induced algebra construction, and both steps have left adjoints. More directly, a map of R-algebras $L_\rho B \to X$ is the same as a function $f : S \times B \to X$ equalizing the two maps in (34). Its transposition $\hat{f} : B \to X^S$ satisfies exactly the equation for a T-algebra map $B \to I_\rho X$. This proves the displayed adjunction and its presentation. For free $B = F_T(X)$,

$$\begin{aligned} \mathrm{Alg}(R)(L_\rho F_T X, Y) &\cong \mathrm{Alg}(T)(F_T X, I_\rho Y) \\ &\cong \mathbf{Set}(X, Y^S) \cong \mathbf{Set}(S \times X, Y), \end{aligned}$$

so Yoneda yields (35). Tracing the free generators identifies the unit with the runner evaluation. No accessibility hypothesis is needed for this fixed-Set-runner adjunction; such hypotheses matter for other existence statements, notably universal Sweedler homs in chosen ambient categories. □

Remark 9.2 (The tensor need not preserve distinctions). A source algebra can impose equations that collapse residual outcomes: the object $L_\rho B$ is not, by itself, a faithful record of winning versus losing. The game semantics is controlled by the runner, an outcome handler, and the chosen set of tests. The unit at a free algebra also simply recovers ρ, so the adjunction is not offered as a new description of the EF strategy encoding.

9.1 Where the adjunction contributes to counting

Take the List runner of Section 6 and the residual List-algebra $(\mathbb{N}, +, 0)$. The induced algebra $I_\rho(\mathbb{N}) = \mathbb{N}^S$ interprets asks as the weighted sums (25). In particular it satisfies the interchange equations (26). Thus it receives homomorphisms from the quotient of the free challenge theory that imposes independence of differently named vertex questions. The adjunction transposes such interpretations into R-algebra morphisms out of $L_\rho B$. In this example it is the *handler* which justifies commutation, not literal equality of ordered lists of executions.

This is a substantive use of the algebraic side: non-free algebras and their equations can identify implementation details invisible to the chosen observations. It also suggests a refined counting semantics using a commutative multiset residual, though the finite-list example has the advantage of being explicitly polynomial.

10 Dependent games and possible composition theorems

There are three natural directions for extending the proved calculations, and it is important not to mistake them for results already obtained.

Uniformity and dependent typing. The polynomial $p_{A,B}$ depends on the particular pair of structures. A theory uniform in a signature σ and resource bound k should account for worlds of partial plays and for answer types that depend on which side was challenged. The many-sorted, slice-polynomial version belongs in our dependent Sweedler programme. Legality should be tracked by witnesses or subobjects, as in the arrow-topos certified-runner examples. The game comonad $\mathbb{E}_k$ supplies a natural source of history-indexing data, but it does not automatically define that dependent theory.

Canonical arenas and alternation. A chosen Duplicator strategy is a section selecting one response at each permitted history. A more canonical game *arena* would retain both Spoiler's universal branching and Duplicator's existential branching, without choosing either. Monotone neighbourhood or game-logic residual monads are candidates for interpreting alternation. The basic finite-powerset residual alone does not designate which player controls which branching. If an arena semantics exists, a winning strategy should become pointing or selection data on it. No canonical construction is claimed here.

Comparison with concurrent game semantics. Montacute and Winskel [14] construct a systematic origin for one-sided game comonads using concurrent games on event structures. Their categories of games and strategies arise from comonads in a bicategory of game schemas; the framework explicitly records causal dependence, independence, and conflict, and also treats two-sided games. This is closer to a canonical operational *arena* than the selected response policies considered here. Our current contribution is different: polynomial free-monad patterns, stateful residual implementations, and finite observation via handlers and bounded temporal comodels. A meaningful comparison must explain how guarded response subcoalgebras, residual effects, and the proposed $\triangleleft/\otimes$ syntax translate into their event-structure arenas and strategies. No such comparison is established here.

Composition. Jakl–Marsden–Shah develop categorical composition methods of Feferman–Vaught–Mostowski type [12]. The strongest possible transfer to runners would compare their logical composition conditions with composition of residual experimental semantics, respecting resource grades and separating observations. The sum–product factorization of independent forest subtrees in Theorem 7.1 is an elementary target case, but neither that factorization nor the monoidal structure of Run by itself establishes a general composition theorem.

11 What has been established?

Status	Statement
Established	The bounded observable behaviour of a pointed Mealy comodel is a map $L_k^+ A \to B$; Jakl's coKleisli legality is a relational lifting (Theorem 3.1).
Established / encoding	A certified residual runner on game histories encodes a Duplicator strategy, and a finite losing EF position admits an adaptive separating term (Proposition 5.1).
Established	A List residual runner enumerates homomorphisms; its additive handler returns $\lvert \mathrm{Hom}(P, A) \rvert$, and the Bag quotient validates order interchange before handling (Theorem 6.1, Proposition 6.3).
Established	A tree-depth forest cover yields a sum–product circuit with at most k nested sums (Theorem 7.1).
Established	Bounded response behaviours form the final guarded Mealy coalgebra; legal one-sided coKleisli maps form a canonical subcoalgebra. Legality of a responder is equivalent to factorization of its behaviour map (Theorem 4.3, Corollary 4.4).
Established	Finite adaptive game tests have compositional cylinder guards and separate bounded response behaviours (Proposition 4.6).
Established	Histories and future response tests satisfy the Hankel identity $e_v \sigma_u = e_{uv}$ and yield minimal Mealy realization (Theorem 8.1).
Established	Every fixed Set-runner gives the relative tensor adjunction $L_\rho \dashv I_\rho$ (Theorem 9.1).
Open	Internalize bounded-height sum–product tests using $\triangleleft/\otimes$ and analyze analytic Bag residues; compare certified runners with Montacute–Winskel event structure games; construct a uniform dependent-game theory and a genuinely compositional comparison with Jakl.

Conclusion. The first contact between finite-model games and Sweedler theory is a behaviour map, not an identification of comonads. Spivak's polynomial dual realizes a continuing Duplicator response machine; Jakl's lifted history comonad determines when its bounded behaviour is relationally legal. Changing the residue from exceptions to lists turns the same idea of executing an experiment into a homomorphism-counting calculation. The forest recurrence shows where logical resource bounds require more than a sequential free monad. The resulting history–test pairing then fits the two geometries of execution and observation developed in the previous interlude. The temporal refinement makes bounded histories and legal one-sided strategies functorial in the remaining round budget, while guarded finite tests provide a direct finite-depth separation theorem. In addition, the canonical subcoalgebra of legal one-sided behaviours makes the safety/legality certificate a precise factorization condition. The duoidal interaction of sequential ancestry and independent subtrees, and the comparison with Montacute–Winskel's event-structure games, remain open structural tasks alongside a uniform dependent Sweedler treatment of two-sided games and residual temporal Chu measurement.

References

[1] *Sweedler Runners for Temporal Systems, I: From Graded Operational Syntax to Temporal Behaviour*, companion research draft, 16 October 2026.

[2] Tomáš Jakl, *Categorical Finite Model Theory: Game Comonads and Arboreal Categories*, habilitation thesis, 2026. https://tomas.jakl.one/posts/habilitation.

[3] S. Abramsky, A. Dawar, and P. Wang, *The pebbling comonad in finite model theory*, LiCS 2017.

[4] S. Abramsky and N. Shah, *Relating Structure and Power: Comonadic Semantics for Computational Resources*, Journal of Logic and Computation 31(6) (2021), 1390–1428. https://arxiv.org/abs/1806.09031.

[5] S. Libkind and D. I. Spivak, *Pattern Runs on Matter: The Free Monad Monad as a Module over the Cofree Comonad Comonad*, EPTCS 429 (2025), 1–28. https://arxiv.org/abs/2404.16321.

[6] G. Plotkin and J. Power, *Tensors of comodels and models for operational semantics*, Electron. Notes Theor. Comput. Sci. 218 (2008), 295–311.

[7] T. Uustalu, *Stateful Runners of Effectful Computations*, Electron. Notes Theor. Comput. Sci. 319 (2015), 403–421. https://doi.org/10.1016/j.entcs.2015.12.024.

[8] D. Ahman and A. Bauer, *Runners in Action*, ESOP 2020, LNCS 12075, 29–55. https://doi.org/10.1007/978-3-030-44914-8_2.

[9] D. McDermott, E. Rivas, and T. Uustalu, *Sweedler theory of monads*, FoSSaCS 2022, LNCS 13242, 428–448.

[10] A. Dawar, T. Jakl, and L. Reggio, *Lovász-Type Theorems and Game Comonads*, LiCS 2021. https://doi.org/10.1109/LICS52264.2021.9470609.

[11] I. Adler, E. Fluck, T. Seppelt, and G. L. Spitzer, *Going deep and going wide: Counting logic and homomorphism indistinguishability over graphs of bounded treedepth and treewidth*, Logical Methods in Computer Science **22**(2), article 33 (2026). https://doi.org/10.46298/lmcs-22(2:33)2026.

[12] T. Jakl, D. Marsden, and N. Shah, *A categorical account of composition methods in logic*, Logical Methods in Computer Science 21(4) (2025), article 10. https://doi.org/10.46298/lmcs-21(4:10)2025.

[13] D. I. Spivak, *The free monad, cofree comonad, and topological space associated to a polynomial*, Topos Institute Berkeley Seminar, June 2022. https://dspivak.net/talks/pdfs/20220614-topos-scotttopology.pdf.

[14] Y. Montacute and G. Winskel, *Concurrent Games over Relational Structures: The Origin of Game Comonads*, LICS 2024. https://arxiv.org/abs/2405.11267.

[15] A. D. Fairbanks, K. Carlson, and D. I. Spivak, *Comonads as Spaces*, arXiv:2607.15091 (2026).

[16] T. Uustalu and V. Vene, *The Essence of Dataflow Programming*, APLAS 2005, LNCS 3780, 2–18.