Getting to Grips with Sweedler Runners for Systems

A conversational tutorial, Part VI.A: Boolean copresheaves
Witnesses, residual runners, and certified wiring

October 2026

Aim. Part V studied safety predicates and assume–guarantee contracts; Part VI developed C-sets and their enriched endofunctor setting. Here we choose the smallest nontrivial cost monoid, $C = \mathbb{B} = \{0 \leq \infty\}$, interpreting 0 as zero cost (truth) and ∞ as unbounded cost. A C-set is then an *arbitrary map of sets*, not necessarily a subobject. We calculate unary comodels, Moore machines and Moore-to-Moore runners, including nonconstant interfaces, and add a short polynomial-lottery example with finite-horizon safety witnesses. Finally, witness-transforming wiring yields an explicit assume–guarantee compositionality argument, while a witnessed good regulator illustrates how certificates evolve under closed-loop dynamics. We use the structural framework developed in Part VI and its appendix; the purpose here is to work with its simplest nontrivial objects, not to revisit its existence hypotheses.

Contents

1 Boolean costs without restricting to predicates **2**

2 Unary dynamics: the first three measurements **3**

3 Moore comodels and behaviour in the arrow topos **3**

4 Moore-to-Moore runners: normal form **4**
4.1 Constant interfaces: an arrow of runners 5
4.2 Nonconstant interfaces: computations carry their own abstraction 6

5 Abstracting commands as well as observations **6**

6 Runner morphisms and horizontal composition **7**
6.1 Two composable nonconstant-interface runners 8

7 Polynomial lotteries and probabilistic witnesses **8**
7.1 A lottery whose tickets have witnesses 9
7.2 A randomised unary runner 9
7.3 A lottery-to-lottery runner that records path weights 9
7.4 Witnesses for a finite-horizon safety bound 10

8 Certified wiring: transforming the witnesses **11**
8.1 Certified wiring is itself a residual-free runner 12
8.2 A complete cascade 12

9 Tight interpretations: what is automatic? 13

10 Good regulators and witnessed belief updating 13
10.1 A regulating relation with evolving witnesses 14
10.2 Belief as a family of arrows 14
10.3 Dependent sum before existential image 15

11 What this special case establishes—and its boundary 16

A The Cayley bridge: from arrows to enriched endofunctors 17

1 Boolean costs without restricting to predicates

Let $\mathbb{B} = (\{0, \infty\}, \leq, +, 0)$ be the ordered commutative monoid with addition given by maximum. Thus $0 + 0 = 0$ and $0 + \infty = \infty + \infty = \infty$. Viewed as a category, it has one nonidentity arrow $0 \to \infty$. Its copresheaf category is

$$\mathcal{E} = \mathbb{B}\text{-}\mathbf{Set} = [\mathbb{B}, \mathbf{Set}] \cong \mathbf{Set}^{\to}.$$

An object is a function $f : A \to B$, and a morphism is a commuting square. In the cost reading of Part VI, $X(0)$ records what is available at zero cost. Reading its elements as witnesses of the elements of $X(\infty)$ therefore arises naturally from the cost grading: an element at stage 0 has an image at stage ∞, while the converse need not hold. Thus at stage 0 we think of A as witnesses or concrete realisations; at the stage ∞ we think of B as the states, observations or inputs being witnessed. A map f need not be injective: distinct witnesses may certify the same observation, and an observation outside $\mathrm{im}(f)$ may have no witness.

The ordinary predicate $P \subseteq B$ is the special case $P \hookrightarrow B$. For general f its image $\mathrm{im}(f) \hookrightarrow B$ is the *truth-valued shadow*, which forgets the fibres $A_b = f^{-1}(b)$ and their maps. Both interpretations will be useful, but we will not impose monicity.

The Day convolution for $+$ on $\mathbb{B}$ is especially simple. For arrows $f : A \to B$ and $k : U \to V$,

$$(f \star k)(0) = A \times U, \qquad (f \star k)(\infty) = B \times V, \qquad f \star k = f \times k.$$

Indeed $a + b \leq 0$ forces $a = b = 0$, while (∞, ∞) is terminal in the coend indexing category at ∞. The unit is $1 \xrightarrow{\mathrm{id}} 1$. Thus Day convolution here coincides with the cartesian structure of the arrow topos. Its internal hom, however, is *not* pointwise. For $f : A \to B$ and $k : U \to V$ it is

$$(k^f)(0) = U^A \times_{V^A} V^B \longrightarrow (k^f)(\infty) = V^B, \qquad (u, v) \longmapsto v,$$

where the pullback equation is $ku = vf$. At stage 0, an internal function is itself a compatible square; at stage ∞ one remembers only its abstract component. This formula is responsible for the genuinely enriched phenomena below.

The examples begin directly with dynamics on witness arrows. Their connection with the enriched endofunctor category of Part VI, including the Cayley state-passing construction and the choice of arities, is explained in Appendix A. For now we can read the calculations as explicit machines, operational translations, and maps between their witnesses.

Scope of the foundation. We work with $\mathcal{E}$-enriched accessible endofunctors in the manner of Part VI, and with the indicated internal algebraic theories and their runners. The general duoidal and measuring-comonoid hypotheses are discussed in Part VI and its appendix; the expanded treatment of systems of arities belongs to the separate Arities Part. No additional general-universe theorem is claimed here.

2 Unary dynamics: the first three measurements

Write $N = \mathbb{N} \xrightarrow{\mathrm{id}} \mathbb{N}$ for the natural-numbers object of $\mathcal{E}$ and let $Ny = N \times -$ denote the free unary-operation monad. A comodel on $S = (A \xrightarrow{f} B)$ is an endomorphism $d : S \to S$, namely

$$\begin{array}{ccc} A & \xrightarrow{g} & A \\ {\scriptstyle f}\downarrow & & \downarrow{\scriptstyle f} \\ B & \xrightarrow[h]{} & B \end{array} \qquad fg = hf.$$

The image $\mathrm{im}(f)$ is invariant under h, but g also transports the fibres $A_b \to A_{h(b)}$. If $f(a_0) = f(a_1) = s$, $h(s) = s$, and g exchanges a_0, a_1, the image predicate only says that s is invariant; the arrow remembers the alternating witnesses.

The three elementary measuring comonads have pointwise descriptions on arrows, because the relevant arities N are constant:

$$\begin{aligned} \{Ny, \mathrm{Id}\}(f) &= (A^{\mathbb{N}} \xrightarrow{f^{\mathbb{N}}} B^{\mathbb{N}}), \\ \{Ny, Ny\}(f) &= ((A \times \mathbb{N})^{\mathbb{N}} \xrightarrow{(f \times \mathrm{id})^{\mathbb{N}}} (B \times \mathbb{N})^{\mathbb{N}}), \\ \{\mathrm{Id}, Ny\}(f) &= f. \end{aligned}$$

The first is the stream comonad: its coalgebras are commuting endomorphisms (g, h). In the second, a coalgebra is such an endomorphism together with a step-count map $k : S \to N$. Writing its two components as $k_A : A \to \mathbb{N}$ and $k_B : B \to \mathbb{N}$, we have

$$k_A = k_B f, \qquad K_n^A(a) = \sum_{r=0}^{n-1} k_A(g^r a) = K_n^B(f(a)).$$

Each source tick executes $k_A(a)$ target ticks and updates the residual state by g; compatible witnesses may evolve differently but must have the same abstract step-count behaviour. The final case is the identity comonad: the source has no effect to translate, so every arrow admits the unique runner into Ny.

3 Moore comodels and behaviour in the arrow topos

Let the input and observation objects themselves be arrows

$$I = (I_0 \xrightarrow{j} I_\infty), \qquad O = (O_0 \xrightarrow{q} O_\infty).$$

By $T_{\mathrm{M}}(I, O)$ we mean the internal algebraic theory freely adjoining an I-indexed family of unary updates to the reader theory for O. We impose the reader equations but *no* equations commuting

reading with updating. This is the theory whose comodels are ordinary Moore machines internally to $\mathcal{E}$:

$$S \longrightarrow O \times S^I, \quad \text{equivalently} \quad \delta : S \times I \longrightarrow S, \quad o : S \longrightarrow O.$$

For $S = (A \xrightarrow{f} B)$ this amounts to $\delta_0 : A \times I_0 \to A$, $\delta_\infty : B \times I_\infty \to B$, $o_0 : A \to O_0$, and $o_\infty : B \to O_\infty$, satisfying

$$f\delta_0 = \delta_\infty(f \times j), \qquad qo_0 = o_\infty f. \tag{1}$$

If two witnesses a, a' map to the same abstract state, their concrete observations need only have the same *abstract* observation:

$$f(a) = f(a') \quad \Longrightarrow \quad q(o_0(a)) = q(o_0(a')).$$

For example let $A = \{a_0, a_1\}$, $B = \{s\}$, let a single tick exchange a_0 and a_1 while fixing s, and let

$$O = (\{\mathsf{red}, \mathsf{blue}\} \longrightarrow \{\mathsf{ok}\}).$$

Read red at a_0 and blue at a_1. The abstract behaviour is constantly ok, while the concrete traces alternate red/blue. This is behavioural abstraction, not an invariant subset.

Internally, the final Moore behaviour object is

$$\mathrm{Beh}_{I,O} = O^{I^*},$$

where I^* is the free internal monoid. In this presheaf topos I^* is computed pointwise, giving $I_0^* \to I_\infty^*$. Hence

$$\begin{aligned} \mathrm{Beh}_{I,O}(0) &= \{(\beta_0, \beta_\infty) \mid \beta_0 : I_0^* \to O_0,\ \beta_\infty : I_\infty^* \to O_\infty,\ q\beta_0 = \beta_\infty j^*\}, \\ \mathrm{Beh}_{I,O}(\infty) &= O_\infty^{I_\infty^*}. \end{aligned}$$

The structure map forgets the concrete component. In particular, the behaviour map of a Moore machine is an arrow whose upper component sends concrete states to compatible concrete/abstract behaviours. The compatibility equation on words is an immediate iterated consequence of (1).

4 Moore-to-Moore runners: normal form

Now put $T = T_{\mathrm{M}}(I, O)$ and $R = T_{\mathrm{M}}(I', O')$. A runner with residual object S is a monad morphism

$$T \longrightarrow \mathrm{St}_S(R), \qquad \mathrm{St}_S(R)(X) = (R(S \times X))^S,$$

where all products and powers are internal to $\mathcal{E}$. We first calculate over sets, for nonempty observation sets, then internalise.

A source unary update $i \in I$ may be realised by an arbitrary target program returning a new residual state:

$$u : S \times I \longrightarrow R(S). \tag{2}$$

This target program can make updates, read its target observation, branch, and return a residual state. By contrast, the source observation get satisfies the reader laws. A priori its translation has type $S \to R(S \times O)$, but those equations force it to be read-only and state-preserving.

Proposition 4.1 (Moore-runner normal form)**.** *Over* **Set***, for the free Moore theory just specified with nonempty target observation set $O\prime$, a runner on S is determined by, and determines, maps*

$$\phi : S \times O' \longrightarrow O, \qquad u : S \times I \longrightarrow R(S). \tag{3}$$

For $\mathcal{E} = \mathbf{Set}^{\to}$ the same classification is expected under the normal-form/gluing condition discussed below, but is not proved here in full generality. The internal constructions used below do not depend on the converse classification.

Normal-form argument. The source updates have no relations other than functorial substitution, so their images are unrestricted target programs of the displayed type. For the source read operation, apply the equations saying that a read with identical continuations is the continuation itself and that consecutive reads collapse to one read. The Set argument can be tested in the free target theory by interpreting terms as functions from target-update histories and observation responses to residual returns: update words remain recorded, so the first reader equation rules out any nontrivial updates in the translated read. The unit equation likewise forces the residual return to be unchanged. With updates excluded, reader normalisation identifies any finite sequence of target reads with a single target read, followed by a selection of an O-continuation determined by the initial residual state and target observation. That selection is precisely ϕ. Conversely, the operation 'read o', continue at $\phi(s, o')$ without changing s' satisfies the reader equations. The free coproduct with unary updates supplies no cross-equations. This gives the Set normal form when the target observation set is nonempty. □

Internalisation checkpoint. The arrow topos is not Boolean, so the preceding Set proof cannot simply be repeated by case distinctions on target observations. Evaluation at ∞ yields the usual Set free-Moore monad and hence its normal form. At stage 0 an internal term retains a compatible abstract continuation, and one expects an embedding into the product of the concrete Set term model and the abstract term model. Establishing that this embedding is faithful modulo reader equations, and that its normal forms glue, is an additional step. Accordingly, the internal construction from (ϕ, u) is valid without a gluing lemma: interpret the source read as the target read followed by ϕ, retaining the residual state; the reader equations hold by equational reasoning internally, and the source updates are free. The unrestricted internal *exhaustiveness* assertion is the remaining proof obligation.

The normal form is best understood as a description of *effectful implementations*. A source update need not correspond to one target update, or even to a fixed number of them. It may execute a branching target program. We have not equated the Moore *monad* with the polynomial Moore *coalgebra functor* $O \times (-)^I$; their roles are different.

4.1 Constant interfaces: an arrow of runners

Suppose I, O, I', O' are constant arrows $X \xrightarrow{\mathrm{id}} X$ and $S = (A \xrightarrow{f} B)$. Then (3) unpacks as

$$\begin{aligned} &\phi_A : A \times O' \to O, && u_A : A \times I \to R(A), \\ &\phi_B : B \times O' \to O, && u_B : B \times I \to R(B), \end{aligned}$$

with compatibility

$$\phi_A(a, o') = \phi_B(f(a), o'), \qquad R(f)(u_A(a, i)) = u_B(f(a), i). \tag{4}$$

The second equation compares *whole target programs* modulo the target equations, not merely their final residual states. Branches differing only in their returned residual states may merge under $R(f)$; with constant interfaces, branches executing genuinely different target update commands cannot merge, because $R(f)$ leaves commands unchanged. Thus a runner is an arrow between two Set-based Moore runners when the interfaces are constant.

4.2 Nonconstant interfaces: computations carry their own abstraction

For general internal interfaces write

$$S = (A \xrightarrow{f} B), \quad I = (I_0 \xrightarrow{j} I_\infty), \quad O = (O_0 \xrightarrow{q} O_\infty), \quad I' = (I'_0 \xrightarrow{j'} I'_\infty), \quad O' = (O'_0 \xrightarrow{q'} O'_\infty).$$

The observation part of (3) consists of $\phi_0 : A \times O'_0 \to O_0$ and $\phi_\infty : B \times O'_\infty \to O_\infty$ with

$$q\phi_0(a, o') = \phi_\infty(f(a), q'(o')). \tag{5}$$

For updates we instead have $u_0 : A \times I_0 \to R(S)(0)$ and $u_\infty : B \times I_\infty \to R(S)(\infty)$, with

$$R(S)(0 \to \infty)\, u_0(a, i) = u_\infty(f(a), j(i)). \tag{6}$$

Crucially $R(S)(0)$ cannot generally be replaced by $R_0(A)$: even the reader expression $S^{O'}$ has concrete component

$$A^{O'_0} \times_{B^{O'_0}} B^{O'_\infty}.$$

The concrete branch of an internal target program therefore includes coherent abstract information. This is the point where the arrow topos is doing more than placing two unrelated Set-based runners alongside one another. Branch merging by identification of returned residual states, however, already occurs with constant interfaces. What a nonconstant $I\prime$ additionally permits is the identification of distinct target update commands; a nonconstant $O\prime$ can remove concrete observation alternatives at the abstract stage.

5 Abstracting commands as well as observations

A non-injective residual arrow can merge branches even when the interfaces are constant: if two branches execute the same commands and return a_L or a_R with $f(a_L) = f(a_R)$, their images under $R(f)$ coincide and the reader equation can remove the now-redundant read. The distinctive possibility with a *nonconstant input interface* is that different concrete commands themselves become the same abstract command.

Take residual and source interfaces

$$S = (\{a_L, a_R\} \xrightarrow{f} \{s\}), \qquad I = (1 \xrightarrow{\mathrm{id}} 1), \qquad O = (\{\mathsf{red}, \mathsf{blue}\} \xrightarrow{q} \{\mathsf{ok}\}),$$

with f, q constant. Give the target the observation arrow

$$O' = (\{L, R\} \xrightarrow{q'} \{*\})$$

and the *nonconstant update interface*

$$I' = (\{\mathsf{step}_L, \mathsf{step}_R\} \xrightarrow{j'} \{\mathsf{step}\}), \qquad j'(\mathsf{step}_L) = j'(\mathsf{step}_R) = \mathsf{step}.$$

Put $\phi_0(a, L) = \mathsf{red}$ and $\phi_0(a, R) = \mathsf{blue}$ for both witnesses, with $\phi_\infty(s, *) = \mathsf{ok}$. This satisfies the observation-compatibility equation (5).

Translate the source update at a concrete witness a by the finite target program

$$\mathsf{get}_{O'};\quad \begin{cases} \mathsf{step}_L;\ \mathrm{return}(a_L), & \text{if } L,\\ \mathsf{step}_R;\ \mathrm{return}(a_R), & \text{if } R. \end{cases}$$

At the abstract stage use $\mathsf{step}; \mathrm{return}(s)$. After applying j', q' and f, the two concrete branches agree as abstract target programs; the read over the singleton then disappears by the reader law. This explicitly satisfies (6). Unlike the constant-interface example, the *distinct update commands* $\mathsf{step}_L, \mathsf{step}_R$ themselves are identified by the interface arrow. In a fixed-command target, $R(f)$ could identify the returned states but could not identify different commands.

The result concerns translations of source operations into target *programs*. A target comodel or handler is needed to execute those programs. Moreover, the concrete term here must be understood with its compatible abstract continuation, as required by the internal exponential; the displayed equations verify that compatibility in this instance.

6 Runner morphisms and horizontal composition

For fixed source and target theories, let (S, ϕ, u) and (S', ϕ', u') be runners. A morphism between them is an internal map $m : S \to S'$ satisfying

$$\phi = \phi'(m \times \mathrm{id}_{O'}), \qquad R(m)u = u'(m \times \mathrm{id}_I). \tag{7}$$

These are the two-dimensional cells between loose arrows with fixed endpoints. They preserve the *programs* attached to source updates as well as the way source observations are read. For a concrete nonidentity example, collapse a_L, a_R to a single witness b over s, keeping ϕ as in Section 5. The distinction between two versions of the update is instructive. In the *single-command variant*, both concrete branches execute the same target command step and return a_L or a_R respectively. After collapsing witnesses, the one-witness update reads L/R, executes step and returns b in either branch. Its continuations now coincide, so the reader equation removes the read, leaving $\mathsf{step}; \mathrm{return}(b)$. In the *two-command example* of Section 5, the one-witness update instead becomes

$$\mathsf{get}_{O'};\quad \begin{cases} \mathsf{step}_L; \mathrm{return}(b), & L,\\ \mathsf{step}_R; \mathrm{return}(b), & R. \end{cases}$$

The read survives: the commands retain the branch distinction that the collapsed witnesses no longer carry. In both versions the collapse map $m : (\{a_L, a_R\} \to \{s\}) \to (\{b\} \to \{s\})$ satisfies (7); the abstract component still identifies the two commands. Thus abstraction of witnesses may remove a read, but need not do so when the subsequent target effects remain distinguishable.

For horizontal composition, let

$$T \xrightarrow[(S)]{(\phi,u)} R, \qquad R \xrightarrow[(Q)]{(\psi,v)} U.$$

The composite residual is $S \times Q$. If the observation interfaces of T, R, U are respectively O, O', O'', its read translation is

$$\chi((s, t), o'') = \phi(s, \psi(t, o'')). \tag{8}$$

Its update translation substitutes the second runner's interpretation into the first runner's R-program, threads the Q-state, and returns both residual states. More precisely, if

$$\widehat{v} : R(S) \times Q \longrightarrow U(S \times Q)$$

is the extension of the second runner to R-computations with returned values in S, then

$$w((s,t),i) = \widehat{v}(u(s,i),t). \tag{9}$$

The composite residual arrow is the product of the two residual arrows; compatibility follows from internality of substitution and products.

6.1 Two composable nonconstant-interface runners

For this compositional calculation, take the single-command variant of the preceding example, so the intermediate update command is step. Extend it by a runner from the intermediate Moore theory to a third one. Give the final target observation arrow

$$O'' = (\{x, y\} \longrightarrow \{\bullet\})$$

and its sole update command the name pulse. The second runner has residual $Q = (\{c_0, c_1\} \to \{t\})$. Translate intermediate read by $x \mapsto L$, $y \mapsto R$ (and $\bullet \mapsto *$). Translate each intermediate step by two pulse commands followed by the residual swap $c_0 \leftrightarrow c_1$; at the abstract level execute two pulses and return t.

Substitution into the first runner gives an explicit composite source update at (a, c); the branch is decided by the target observation read *before* the pulses, not by a new read after them:

$$\mathsf{get}_{O''};\quad \mathsf{pulse};\ \mathsf{pulse};\quad \begin{cases} \mathrm{return}(a_L, \mathrm{swap}(c)), & x, \\ \mathrm{return}(a_R, \mathrm{swap}(c)), & y. \end{cases}$$

Its abstract translation is simply two pulses followed by $\mathrm{return}(s,t)$. The composite observation translation sends x to red, y to blue, and $\bullet$ to ok. This calculation illustrates (8)–(9), including the disappearance of branches under abstraction. It also exhibits the ordinary mechanism of horizontal composition rather than just asserting its existence.

7 Polynomial lotteries and probabilistic witnesses

The witness arrows of this Part can carry random dynamics as well as deterministic ones. Here the polynomial lottery monad from Part II is especially convenient: it retains individual tickets, including repeated labels, rather than identifying all presentations of the same probability distribution. We use lotteries with at least one ticket,

$$R(X) = \coprod_{n \geq 1} \Delta_n \times X^n, \qquad \Delta_n = \{(p_1, \ldots, p_n) \in [0,1]^n \mid \textstyle\sum_i p_i = 1\}.$$

Its unit is the one-ticket lottery and its multiplication flattens nested lotteries, multiplying the outer and inner ticket weights. This is a polynomial monad on **Set**. Writing $\underline{K} = (K \xrightarrow{\mathrm{id}} K)$ for a constant arrow, the corresponding internal polynomial on $\mathcal{E}$ is

$$R_{\mathcal{E}}(X) = \coprod_{n \geq 1} \underline{\Delta_n} \times X^{\underline{n}}, \qquad R_{\mathcal{E}}(A \xrightarrow{f} B) = (R(A) \xrightarrow{R(f)} R(B)).$$

In the second equality we use that exponentials by constant arrows and coproducts are calculated pointwise. The unit and flattening likewise define maps of arrows.

7.1 A lottery whose tickets have witnesses

For $f : A \to B$, let $A_b = f^{-1}(b)$. Over a particular formal lottery $\lambda = (p_1 : b_1, \ldots, p_n : b_n) \in R(B)$, the arrow $R(f)$ has fibre

$$R(f)^{-1}(\lambda) \cong \prod_{i=1}^{n} A_{b_i}. \tag{10}$$

One chooses a concrete witness for each *ticket*, not merely for each distinct outcome. An abstract lottery supported outside $\mathrm{im}(f)$ remains available on the right, even if it has no concrete witness. Because these are formal tickets, a ticket of weight zero still needs a witness for a strict lift; that requirement would disappear after taking the finite-distribution quotient.

A different operation is to choose an entire *lottery* of witnesses in each fibre. Conditional lotteries $\kappa_b \in R(A_b)$ may refine an abstract ticket b to several concrete tickets. Flattening them preserves the induced distribution on B, but generally not the formal lottery: $(\frac{1}{2} : b, \frac{1}{2} : b)$ is not the one-ticket lottery $(1 : b)$. We shall keep these two senses of probabilistic refinement distinct.

7.2 A randomised unary runner

A runner of the free unary theory Ny into $R_{\mathcal{E}}$ on residual arrow $S = (A \xrightarrow{f} B)$ consists of a morphism

$$\rho : S \longrightarrow R_{\mathcal{E}}(S), \qquad R(f)\rho_A = \rho_B f. \tag{11}$$

It interprets one tick as a lottery of successor residual states; the free source has no further equations. For example, let $f(a_0) = f(a_1) = s$, $f(c) = t$, and put

$$\begin{aligned} \rho_A(a_0) &= (\tfrac{1}{3} : a_1, \tfrac{2}{3} : c), \quad \rho_A(a_1) = (\tfrac{1}{3} : a_0, \tfrac{2}{3} : c), \quad \rho_A(c) = (1 : c), \\ \rho_B(s) &= (\tfrac{1}{3} : s, \tfrac{2}{3} : t), \qquad\qquad \rho_B(t) = (1 : t). \end{aligned}$$

The abstract transition forgets the alternation of the two witnesses over s. Two ticks from a_0 produce the three-ticket lottery

$$\rho_A^{(2)}(a_0) = (\tfrac{1}{9} : a_0, \tfrac{2}{9} : c, \tfrac{2}{3} : c),$$

and applying $R(f)$ gives the abstract two-tick lottery. In particular, we do not merge repeated tickets.

The corresponding measuring comonad has a helpful concrete description. Subject to the usual cofree-coalgebra existence hypotheses, the measurement $\{Ny, R_{\mathcal{E}}\}$ is the cofree comonad on the endofunctor $R_{\mathcal{E}}$:

$$\{Ny, R_{\mathcal{E}}\}(X) \cong \nu Z.(X \times R_{\mathcal{E}}(Z)). \tag{12}$$

Its elements are possibly infinite, finitely branching execution trees, with a state label at each node and a probability vector on each node's outgoing tickets. The counit reads the root label and comultiplication records all rooted subtrees. A coalgebra unfolds the one-step map (11). Since this particular polynomial is built from constant arities, the trees and their map of witness labels can be calculated pointwise on $A \to B$.

7.3 A lottery-to-lottery runner that records path weights

There is also a genuinely residual self-runner $R \rightsquigarrow R$. A generic lottery operation is indexed by $p = (p_1, \ldots, p_n) \in \Delta_n$. Translating it with residual set S requires a target lottery in $R(S \times \{1, \ldots, n\})$: a

returned index selects the source continuation. For the ticket-preserving theory, the unit operation must translate to $\eta(s, 1)$; translating a flattened nested lottery must agree with sequentially translating its outer and inner lotteries. These are equalities of formal target lotteries, not only of the resulting probability measures.

A useful family satisfying both laws arises when S has an action of the multiplicative monoid $([0, 1], \cdot, 1)$. Define

$$\theta_p(s) = \big(p_i : (s \cdot p_i, i)\big)_{i=1}^{n}. \tag{13}$$

The unit law follows from $s \cdot 1 = s$. For a nested lottery with inner weights q_{ij}, sequential translation returns residual state $(s \cdot p_i) \cdot q_{ij} = s \cdot (p_i q_{ij})$ on ticket (i, j), exactly the flattened translation. Hence (13) defines a self-runner of the polynomial lottery monad. Taking $S = [0, 1]$, starting from 1, records the product of the probabilities of the selected tickets. Its negative logarithm records the additive information cost

$$-\log(p_1 \cdots p_k) = \sum_{i=1}^{k} -\log p_i, \qquad -\log 0 = \infty.$$

This is a direct, elementary connection with the cost monoids of Part VI; it is not yet an amortised expected-cost argument.

The example also lives in the arrow topos. Give A and B multiplicative actions and let $f : A \to B$ preserve them, $f(a \cdot t) = f(a) \cdot t$. The two instances of (13) then commute with f, yielding one internal self-runner with residual object $(A \to B)$. For example, $A = \{L, R\} \times [0, 1] \to B = [0, 1]$ can retain a witness label while the abstract residual records only the accumulated probability.

7.4 Witnesses for a finite-horizon safety bound

A witnessed lottery of states is not, by itself, a quantitative safety guarantee. Let $P \subseteq S$ be a safe region and let $\rho : S \to R(S)$ be a randomised transition. Define $r_N(s)$ as the probability of visiting an unsafe state at or before time N. Then

$$r_0(s) = \begin{cases} 0 & s \in P, \\ 1 & s \notin P, \end{cases} \qquad r_{N+1}(s) = \begin{cases} 1 & s \notin P, \\ \sum_i p_i r_N(s_i) & s \in P,\ \rho(s) = (p_i : s_i)_i. \end{cases} \tag{14}$$

The execution tree in (12) records the branches needed for this calculation. For fixed N, let W_N contain derivations π of bounds $r_N(s) \le \varepsilon$ and form the witness arrow

$$q_N : W_N \longrightarrow S \times [0, 1], \qquad (s, \varepsilon, \pi) \longmapsto (s, \varepsilon).$$

For a safe state s with $\rho(s) = (p_i : s_i)_i$, proofs of $r_N(s_i) \le \varepsilon_i$ combine into a proof of

$$r_{N+1}(s) \le \sum_i p_i \varepsilon_i. \tag{15}$$

A particularly small example is the symmetric random walk $s \mapsto (\frac{1}{2} : s - 1, \frac{1}{2} : s + 1)$ with safe region $\{0, \dots, 10\}$. At $s = 0$, the terminal bounds $r_0(-1) = 1$ and $r_0(1) = 0$ yield $r_1(0) = \frac{1}{2}$: safety is not invariant, but a nontrivial probabilistic guarantee is certified.

The rule (15) respects lottery substitution because

$$\sum_i p_i \Big(\sum_j q_{ij} \varepsilon_{ij} \Big) = \sum_{i,j} p_i q_{ij} \varepsilon_{ij}.$$

Thus a proof may be combined in stages or after flattening the lottery. This is a *witness-producing* probabilistic compositionality principle. It is not yet a general stochastic assume–guarantee theorem: relating composite failure to component failures additionally requires a certified condition on the wiring, and infinite-horizon bounds require further analytic hypotheses. The following sections show how the deterministic wiring argument transports proof objects; a quantitative version would also transport their risk bounds.

8 Certified wiring: transforming the witnesses

The connection with Part V is not that every arrow *is* a predicate. An arrow $q : G \to O$ can instead mean: G consists of *witnesses that observations are good*. Different witnesses may certify the same output. An assumption $p : H \to I$ similarly records witnesses of admissible inputs. A machine's state certification is $r : P \to S$, with witness-producing maps

$$\widehat{o} : P \longrightarrow G, \qquad \widehat{\delta} : P \times H \longrightarrow P, \qquad q\widehat{o} = or, \quad r\widehat{\delta} = \delta(r \times p). \tag{16}$$

The first constructs a guarantee witness; the second transports state-certification witnesses across admissible transitions. They are morphisms in $\mathbf{Set}^{\to}$; none of r, p, q must be monic.

Take n such machines, with observation spaces O_i, input spaces I_i, and certificates (P_i, H_i, G_i). Write the product state $S = \prod_i S_i$, and suppose a wiring lens has an input part and output part

$$w_{\mathrm{in}} : I \times \prod_i O_i \longrightarrow \prod_i I_i, \qquad w_{\mathrm{out}} : \prod_i O_i \longrightarrow O.$$

For the global contract take $p : H \to I$ and $q : G \to O$. A *witness-transforming certified wiring* supplies

$$\widehat{w}_{\mathrm{in}} : H \times \prod_i G_i \longrightarrow \prod_i H_i, \qquad \widehat{w}_{\mathrm{out}} : \prod_i G_i \longrightarrow G, \tag{17}$$

with commuting equations

$$(\prod_i p_i)\widehat{w}_{\mathrm{in}} = w_{\mathrm{in}}(p \times \prod_i q_i), \tag{18}$$

$$q\widehat{w}_{\mathrm{out}} = w_{\mathrm{out}} \prod_i q_i. \tag{19}$$

The first converts global input and component output witnesses into the component input witnesses required for safe operation. The second converts component guarantees into a *global output guarantee*. Both are essential.

Proposition 8.1 (Witness-level assume–guarantee composition)**.** *Suppose each component is equipped with* (16) *and the wiring with* (17)–(19)*. Then the wired Moore machine is certified on* $P = \prod_i P_i \to \prod_i S_i$ *under* $H \to I$*, and its output is guaranteed by* $G \to O$*.*

Proof. The global observation witness is the composite

$$\widehat{o} = \widehat{w}_{\mathrm{out}} \circ \prod_i \widehat{o}_i : \prod_i P_i \longrightarrow G.$$

For the transition, take $(\boldsymbol{\pi}, h) \in \prod_i P_i \times H$, with $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$, and set

$$(\eta_1, \ldots, \eta_n) = \widehat{w}_{\mathrm{in}}(h, \widehat{o}_1(\pi_1), \ldots, \widehat{o}_n(\pi_n)).$$

Then define

$$\widehat{\delta}_{\mathrm{comp}}(\boldsymbol{\pi}, h) = (\widehat{\delta}_1(\pi_1, \eta_1), \ldots, \widehat{\delta}_n(\pi_n, \eta_n)). \tag{20}$$

Equations (16) and (18) show that (20) lies over the transition of the wired machine; equations (16) and (19) show that $\widehat{o}$ lies over its output. Thus both the assumption-discharge and guarantee-production squares commute. □

The proof produces witnesses, rather than establishing only their existence. Passing to the image of each witness map recovers the familiar implication-valued contract, but forgets which certificate was constructed.

8.1 Certified wiring is itself a residual-free runner

There is a structural reason for Proposition 8.1. Regard each triple of witness arrows

$$r_i : P_i \to S_i, \qquad p_i : H_i \to I_i, \qquad q_i : G_i \to O_i$$

as a Moore machine *internal to* $\mathcal{E}$: equations (16) say precisely that $(\widehat{\delta}_i, \delta_i)$ and $(\widehat{o}_i, o_i)$ are internal morphisms with input object p_i and observation object q_i. These are unconditional comodel structures in $\mathcal{E}$, even though their codomain components express an ordinary contract conditional on having an input witness.

The two wiring squares (17) are likewise the input and output parts of a *lens internal to* $\mathcal{E}$,

$$\mathbf{q} = \prod_i q_i, \quad \mathbf{p} = \prod_i p_i, \qquad w_{\mathrm{in}} : p \times \mathbf{q} \to \mathbf{p}, \quad w_{\mathrm{out}} : \mathbf{q} \to q.$$

Here p and q denote the global input-assumption and output-guarantee arrows, whereas the bold symbols $\mathbf{p}$ and $\mathbf{q}$ denote the products of component arrows. The displayed maps are morphisms of arrows, whose stage-0 components are the hatted witness transformers and whose stage-∞ components are the ordinary wiring maps. The lens acts on the product of component Moore machines to give a composite Moore machine internally to $\mathcal{E}$; its stage-0 transition and output are exactly (20) and $\widehat{o}$.

At the operational level the same lens can be read as a *residual-unit Moore runner*. Its observation translation is $\varphi = w_{\mathrm{out}}$ and the interpretation of a global update with input i is: read the tuple of component observations, compute $w_{\mathrm{in}}(i, \vec{o})$, execute the respective component updates and return the unit residual. The appropriate target is a theory of *independent parallel components*, expressed by their tensor/non-interfering combination when that tensor is available, not their unrestricted coproduct. The product of the component state objects supplies its comodel. Composing this runner with that comodel gives Proposition 8.1. This runner is constructed directly from the internal data, so the unproved exhaustiveness direction of Proposition 4.1 is not needed.

Thus certified wiring is not an unrelated extra construction. It is the internal arrow-topos instance of runner evaluation: the witness-producing composite is the stage-0 behaviour and ordinary wiring is its stage-∞ shadow. The novelty relative to ordinary predicate contracts is that the stage-0 certificate maps need not be monic.

8.2 A complete cascade

For a small instance let the first machine output $O_1 = \{0, 1\}$, with guaranteed-output witnesses

$$G_1 = \{g_0, g_0', g_1\} \xrightarrow{q_1} O_1, \qquad q_1(g_0) = q_1(g_0') = 0, \quad q_1(g_1) = 1.$$

Let the second machine accept $I_2 = \{0, 1\}$, and give its input-admissibility witnesses as

$$H_2 = \{h_0, h_0', h_1\} \xrightarrow{p_2} I_2, \qquad p_2(h_0) = p_2(h_0') = 0, \quad p_2(h_1) = 1.$$

The cascade wiring sends the first observation to the second input. Its assumption-discharge witness map is

$$t : G_1 \longrightarrow H_2, \qquad t(g_0) = h_0, \quad t(g_0') = h_0', \quad t(g_1) = h_1, \qquad p_2 t = q_1.$$

Even though g_0, g_0' certify the same observation 0, they are transmitted as distinct witnesses of the second machine's admissibility. The first input simply passes through from the global input, and the second output becomes the global output. Thus $\widehat{w}_{\mathrm{in}}(\eta, \gamma_1, \gamma_2) = (\eta, t(\gamma_1))$ and $\widehat{w}_{\mathrm{out}}(\gamma_1, \gamma_2) = \gamma_2$ (with the evident identifications of global assumption and guarantee objects). The proof of Proposition 8.1 constructs the composite certificates explicitly.

This is the elementary witness-level counterpart of the compositional assume–guarantee mechanism of Capucci and Myers [2]. Their account uses certified lenses and a symmetric monoidal double-categorical organisation, in a broader setting of generalized Moore machines and tangencies. Our example shows how *maps of witnesses* may inhabit the certified interface objects; it is not a claim that their entire theory has been reconstructed by the arrow topos.

9 Tight interpretations: what is automatic?

An operational wiring interpretation may be represented by a monad morphism

$$w : T \longrightarrow U.$$

For unrelated component operations one may start from the coproduct $T_1 + \cdots + T_n$, but the parallel, non-interfering component wiring above naturally factors through an appropriate tensor or commuting combination U of the component theories. This distinction is operational: coproduct alone does not impose the equations expressing independence of component actions. A U-comodel restricts along w to a T-comodel, so satisfaction of the *operational equations* of T is automatic. If a state-certificate arrow $r : P \to S$ is already a morphism of U-comodels, it remains one after restriction along w.

But the local contracts above are *conditional as ordinary Set-level contracts*: the component machines need only preserve their state witnesses under admissible input witnesses. (Their witness-bearing versions are unconditional Moore comodel structures in $\mathcal{E}$, as in Section 8.1.) A general U-comodel need not preserve those witnesses for every operation, so an arbitrary tight interpretation does not imply Proposition 8.1. The wiring must have the lifts $\widehat{w}_{\mathrm{in}}$ and $\widehat{w}_{\mathrm{out}}$ (or equivalent encoded structure). In the arrow topos, the equations asserting their existence are exactly equations for morphisms of witness arrows. If we construct the operational theories and their interpretation *with those witness interfaces built in*, a tight morphism in that certified setting enforces the relevant compatibility.

This gives a precise division of labour: tightness supplies well-defined *operational composition*; certified lifting supplies *assumption discharge and global guarantee production*. Neither side substitutes for the other.

10 Good regulators and witnessed belief updating

Part V interpreted a good-regulator condition through a possibilistic belief map and the predicate adjunction $\exists_f \dashv f^*$. The Boolean copresheaf viewpoint suggests a proof-relevant refinement: retain

the witnesses presenting a regulating relation and ask that the closed-loop dynamics *update those witnesses*. This is a refinement of the earlier construction, not a new proof of a good-regulator theorem without additional hypotheses.

10.1 A regulating relation with evolving witnesses

Let X and Y be agent and environment state sets, and let

$$h : X \times Y \longrightarrow X \times Y$$

be the closed-loop one-step dynamics obtained from their wiring. The ordinary regulating relation is an invariant subset $R \subseteq X \times Y$ lying within a prescribed good region G. Instead present it by a possibly non-injective arrow

$$r : W \longrightarrow X \times Y.$$

A *witnessed invariant regulator* has an endomorphism $d : W \to W$ making the square commute:

$$\begin{array}{ccc} W & \xrightarrow{\ d\ } & W \\ {\scriptstyle r}\downarrow & & \downarrow{\scriptstyle r} \\ X \times Y & \xrightarrow[\ h\ ]{} & X \times Y. \end{array} \qquad rd = hr.$$

Consequently $\mathrm{im}(r)$ is invariant under h. But d records more than invariant membership: it tells us which certificate survives, or how the certificate is transformed, after one step. As in the unary example, (r, d, h) is a unary dynamical system in $\mathcal{E} = \mathbf{Set}^{\to}$.

We can also retain *witnesses of goodness*. Let $q : K \to X \times Y$ present the normative or good-region specification. Require a map $c : W \to K$ with $qc = r$. Thus regulating witnesses explicitly produce goodness witnesses. If the intended objective is an ordinary subset $G \hookrightarrow X \times Y$, we may choose $K = G$; for a general q, its image is the underlying truth-valued specification. Neither r nor q is required to be monic.

10.2 Belief as a family of arrows

Write $r = (r_X, r_Y)$. For each $x \in X$, form the fibre $W_x = r_X^{-1}(x)$ and its environment-state map

$$b_x : W_x \longrightarrow Y, \qquad b_x(w) = r_Y(w).$$

This is a *witnessed possibilistic belief* of the agent at x. Its image is exactly the ordinary belief set

$$\psi(x) = \mathrm{im}(b_x) = \{y \in Y \mid (x, y) \in \mathrm{im}(r)\}.$$

Distinct witnesses in W_x may support the same environment state. Two regulators may therefore have identical ψ but different witness sets and update dynamics.

For a witness $w \in W_x$, write $h(x, b_x(w)) = (x', y')$. The commuting square gives

$$d(w) \in W_{x'}, \qquad b_{x'}(d(w)) = y'.$$

Thus d transports each belief witness to a witness for the actual successor environment state. In general this is *not* a single map $W_x \to W_{x'}$ for one fixed x': the next agent state may depend on the environment state represented by w. The correct structure is the global commuting square, or equivalently fibrewise transport over the induced state transition. Taking images recovers forward consistency of the possibilistic belief relation.

10.3 Dependent sum before existential image

For any function $f : Z \to Z'$, reindexing by pullback has the slice-category left adjoint

$$\Sigma_f : \mathbf{Set}/Z \rightleftarrows \mathbf{Set}/Z' : f^*, \qquad \Sigma_f(P \xrightarrow{p} Z) = (P \xrightarrow{fp} Z').$$

Explicitly,

$$\mathrm{Hom}_{\mathbf{Set}/Z'}(\Sigma_f P, Q) \cong \mathrm{Hom}_{\mathbf{Set}/Z}(P, f^*Q). \tag{21}$$

On the left, a morphism supplies, for each witness $p_0 \in P$, a witness in Q lying over $f(p(p_0))$. The right side describes the same witness transformation over Z. Thus Σ_f retains the witness object rather than replacing it with a subset. Passing to images yields the predicate adjunction $\exists_f \dashv f^*$ used in Part V. More precisely, the image of $\Sigma_f P$ is $\exists_f(\mathrm{im}(p))$, while pulling back a monomorphism agrees with the ordinary inverse-image predicate.

For the regulator itself, take $Z = Z' = X \times Y$ and $f = h$. A morphism $\Sigma_h r \to r$ in $\mathbf{Set}/(X \times Y)$ is exactly a map $d : W \to W$ with $rd = hr$; by (21) this is equivalently a morphism $r \to h^*r$. Taking images gives $\exists_h R \subseteq R$, or equivalently $R \subseteq h^{-1}R$, which is Part V's invariance condition. Likewise $c : W \to K$ with $qc = r$ is a morphism $r \to q$ in the same slice. In **Set** images are pullback-stable: for any arrow $Q \to Z'$ one has $\mathrm{im}(h^*Q) = h^{-1}(\mathrm{im}\, Q)$. Consequently the image reflection from witness arrows to subobjects respects both sides of the dependent-sum/reindexing adjunction, recovering $\exists_h \dashv h^{-1}$ as its predicate shadow.

The relation between the two versions is therefore not metaphorical: existential image is the truth-valued shadow of dependent sum. However, an image-level invariant relation does not specify a *particular* witness update d or goodness map c. Such maps may have to be chosen, and may be unavailable when one requires computability, continuity, naturality or constructive uniformity. The proof-relevant conditions are genuine additional data.

The regulator lesson. Assume–guarantee wiring in the preceding section transforms certificates *across components*. A witnessed regulator transforms certificates *through time*. Both operations are represented by ordinary maps in the Boolean arrow topos. The familiar regulating relation and possibilistic belief map reappear on taking images, but the witness dynamics disappears.

There is a second distinction, between *product* and *joint* certificates. Proposition 8.1 constructs the product arrow $\prod_i P_i \to \prod_i S_i$: it certifies independent local witnesses with their wired evolution. A good regulator instead comes with an arbitrary joint arrow $r : W \to X \times Y$. Its image can encode a correlation between agent and environment states (for example, $y = m(x)$) that is not generally a product of subsets. Thus the product-certificate construction alone does not supply a regulator's internal model or joint relation. Such a relation might be added as a further certified subobject or witness object and preserved by suitable feedback wiring, but it is *additional structure*, not a consequence of the product certificate. This clarifies why the good-regulator example complements, rather than simply repeats, assume–guarantee composition.

The closed-loop map h may itself arise by restricting a Moore–Mealy coproduct theory along a tight tick interpretation, as in Part III. Such a tight map supplies the operational closed loop; the lift d and the goodness transformer c remain substantive certification data unless they have been encoded in a certified interpretation. This is the same qualification encountered for assume–guarantee wiring above.

11 What this special case establishes—and its boundary

The move from 2-valued predicates to arbitrary arrows is not merely a technical trick for obtaining a convenient monoidal topos. It changes the objects we can describe. A certified state may have several proofs; abstraction may identify several implementation states; a translated update may branch concretely even while its abstract program does not; a wiring diagram may compute a new witness from existing witnesses; and lottery branches may carry distinct probabilistic refinements and weighted safety proofs. The image reflector $f \mapsto \mathrm{im}(f) \hookrightarrow B$ gives back truth-valued semantics while forgetting that operational and proof-theoretic information.

There is a further, independent boundary: the product certificates generated by ordinary component wiring do not automatically give the joint witness objects needed to express a regulating relation between agent and environment. Both this need for correlated witnesses and the contextual input families below suggest investigating indexed structures, but they are distinct questions.

The results here remain within *fixed internal interfaces*. Even a nonconstant input arrow $I_0 \to I_\infty$ is not the same as a family of available input types indexed by the current state. Recall Part V's walk on $\mathbb{Z}$, with safe region $\{0, \dots, 10\}$ and contextual admissibility $0 \le s + a \le 10$. Its fibrewise input space is naturally $U \to S$, with $U_s = \{a \mid 0 \le s + a \le 10\}$. An ordinary Moore transition $S \times I \to S$ remains total on a fixed input object. Treating $U \to S$ *as the operational interface* requires a dependent or fibred extension of the theory, not just a new state arrow.

That question can now be pursued separately. For the moment the Boolean copresheaf case has done its job: it shows concretely how the C-set approach of Part VI retains witnesses and abstraction structure, how this enters both comodels and genuinely residual runners, how polynomial lotteries combine probabilistic witnesses and finite-horizon risk bounds, how certified wiring produces the witnesses of composite guarantees, and how witnessed regulators update certificates through time.

Aberlé's compositional verification framework [1] provides a closely related, independently developed proof-relevant treatment of program specifications. Its dependent polynomials are polynomial endofunctors on the arrow category over ordinary polynomial interfaces, and its dependent free monad transports verification witnesses through program substitution. Our Boolean-copresheaf construction shares this witness-transforming interpretation, while situating it in an enriched Sweedler setting of monads, comodels and residual runners. It would be very useful to explore the commonalities and differences between these approaches.

References

[1] C.B. Aberlé *Compositional Program Verification with Polynomial Functors in Dependent Type Theory*, 2026, arXiv:2604.01303

[2] Matteo Capucci and David Jaz Myers, *Compositionality of Lyapunov functions via assume-guarantee reasoning*, 2026, arXiv:2604.03017.

[3] Rory B. B. Lucyshyn-Wright, *Enriched algebraic theories and monads for a system of arities*, Theory and Applications of Categories **31** (2016), 101–137. (See the arities discussion in the main tutorial.)

A The Cayley bridge: from arrows to enriched endofunctors

The passage from arrows of sets to runners is not simply the passage to monads *in* the topos. Following Part VI, put

$$\mathcal{E} = [\mathbb{B}, \mathbf{Set}] \cong \mathbf{Set}^{\to}, \qquad \mathcal{D}_{\mathbb{B}} = \mathcal{E}\text{-}[\mathcal{E}, \mathcal{E}]_{\mathrm{acc}}.$$

The objects of $\mathcal{D}_{\mathbb{B}}$ are accessible $\mathcal{E}$-enriched endofunctors of $\mathcal{E}$ (equivalently, in this tensored setting, accessible endofunctors equipped with coherent tensorial strength). Its two relevant products are *composition* and the enriched convolution product supplied by the monoidal structure of $\mathcal{E}$. Although Day convolution on $\mathcal{E}$ is cartesian, convolution of *endofunctors* is not thereby pointwise cartesian product. We use the duoidal and measuring framework of Part VI, rather than reconstructing its hypotheses here.

There is a useful distinction concerning the choice of arities. The representables for $\mathbb{B}$ are just

$$y(0) = \mathbb{B}(0, -) = (1 \xrightarrow{\mathrm{id}} 1), \qquad y(\infty) = \mathbb{B}(\infty, -) = (\varnothing \to 1).$$

They are tensor-closed and provide the especially economical starting point discussed in Part VI. But choosing *only* these arities restricts the permitted endofunctors: the internal reader $(-)^O$, for example, need not be representable-ary when O has more than one outcome. The Moore calculations in the main text therefore take place in an enriched accessible endofunctor setting large enough to contain the stated internal Moore monads, as provided for by the broader arities approach. We refer to the separate Arities Part for the enlargement and its hypotheses; we neither identify the representable-ary subcategory with all of $\mathcal{D}_{\mathbb{B}}$ nor prove a new closure theorem here.

The internal theories in this Part determine monads in $\mathcal{D}_{\mathbb{B}}$: the unary monad $Ny = N \times -$, and the Moore monad $T_{\mathrm{M}}(I, O)$ generated by internal updates and the internal reader theory. What turns their operations into runners is the *enriched state-passing* construction. For a residual arrow $S \in \mathcal{E}$ and target monad R, define

$$\mathrm{St}_S(R)(X) = \big(R(S \times X)\big)^S. \tag{22}$$

The unit and multiplication use the unit and multiplication of R and thread the S-state through successive computations. A runner $T \rightsquigarrow R$ with residual S is specified by an *enriched* monad map

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

Under the measuring representation, these are the coalgebra structures $S \to \{T, R\}(S)$; this last equivalence uses the Sweedler construction established under the hypotheses discussed in Part VI.

Here is the point at which the arrow-topos enrichment enters the formula. Write $S = (A \xrightarrow{f} B)$ and $K = R(S \times X) = (K_0 \xrightarrow{k} K_\infty)$. Then

$$\big(\mathrm{St}_S(R)(X)\big)_0 = K_0^A \times_{K_\infty^A} K_\infty^B, \qquad \big(\mathrm{St}_S(R)(X)\big)_\infty = K_\infty^B.$$

Thus a concrete-stage state-passing computation is *not* merely a map $A \to K_0$: it is a pair $(u : A \to K_0, v : B \to K_\infty)$ satisfying $ku = vf$. Its abstract component v is part of the same internal computation. In particular, the commuting equations later displayed for unary and Moore runners are supplied by the hom-object, not appended as external simulation axioms.

For a first check, take $T = Ny$ and $R = \mathrm{Id}$. Interpreting the single unary generator in $\mathrm{St}_S(\mathrm{Id})$ amounts to an internal map $S \to S$, hence to $g : A \to A$ and $h : B \to B$ with $fg = hf$. Taking

$R = Ny$ instead permits an internal step count $k : S \to N$ alongside this update, as illustrated in the unary calculation of Section 2. For Moore source and target, the very same bridge yields the internal observation and effectful-update translations calculated in Sections 4–6.

This is the specific sense in which these examples live in the Sweedler universe $\mathcal{D}_{\mathbb{B}}$: we calculate its measuring *coalgebras* through the state-passing universal property, not the entire measuring comonad by an independent convolution calculation. The structural existence and arities arguments remain in Part VI and its appendix.