Must and May Verification by Sweedler Runners

Two proof-relevant liftings of finite nondeterminism to the arrow topos

Research sequel for the categorical systems programme

12 October 2026

Abstract

We show that varying the *target* of a certified Sweedler runner separates must and may verification while leaving the source specification unchanged. For the finite nonempty-powerset monad $\mathcal{N} = \mathcal{P}^{+}_{\mathrm{fin}}$ on sets, we construct two monad liftings to $\mathbf{Set}^{\to}$: the pointwise finite-powerset lifting, whose upper computation covers every ordinary outcome, and an existential lifting, whose upper computation chooses one certified outcome. For an Aberlé-style certified free polynomial theory $\widehat{T}$ we calculate both runner categories, using free-monad extension and arrow-category Yoneda, as coalgebra categories for explicit endofunctors $\widehat{F}_{\mathsf{must}}$ and $\widehat{F}_{\mathsf{may}}$. The lower transition is independent of the supplied precondition witness. Verified implementations act on both categories by restriction, and their safe supports yield two distinct greatest-fixed-point viability operators. A small branching system separates the modalities, while a sums-of-two-squares example exhibits multiple genuine certificates, made behaviourally observable by a certificate-returning command. Neither lifting preserves arbitrary cartesian reindexing, even though their proof-irrelevant predicate images do. We explain why extensional finite powerset is preferable here to the polynomial monad of nonempty lists, distinguish simulation from outcome-covering certification, and identify the scope of the comparison with fibrational weakest-precondition liftings.

Contents

1 The question, and why finite powerset rather than nonempty lists **2**

2 The certified source theory, held fixed **2**

3 Two proof-relevant monad liftings **3**

4 Reindexing certificates and predicates **4**

5 Sweedler runner calculations **5**

6 Compositionality under verified program interpretations **7**

7 Certified viability: two greatest fixed points **8**

8 Two worked machines **9**
8.1 A safe branch and an unsafe branch 9
8.2 Proof relevance: staying or doubling a sum of squares 9

9 What changes with the nonempty-list monad? **10**

10 Allowing deadlock: partial correctness versus possible success **10**

11 Scope and next questions **11**

1 The question, and why finite powerset rather than nonempty lists

The companion note *Certified Mealy Machines as Sweedler Comodels* identifies the set-level verified Mealy constructions of Aberlé [1] with runners of a free polynomial monad on $\mathbf{Set}^{\to}$. It uses the identity target, so an operational command has a unique response and successor. The point of this sequel is to *hold the certified source theory fixed and vary its target monad.*

For finite nondeterminism we choose

$$\mathcal{N}X = \mathcal{P}^{+}_{\mathrm{fin}}(X) = \{U \subseteq X : 0 < |U| < \infty\}, \qquad \eta(x) = \{x\}, \quad \mu(\mathcal{U}) = \bigcup \mathcal{U}. \tag{1}$$

There are three reasons to prefer $\mathcal{P}^{+}_{\mathrm{fin}}$ to List^{+} here. First, ordinary nondeterminism is *extensional*: permuting outcomes or repeating an outcome changes nothing. The powerset monad enforces these equations, whereas lists retain order and multiplicity. Second, the absence of an empty outcome keeps failure or deadlock separate from the must/may distinction. Third, $\mathcal{P}^{+}_{\mathrm{fin}}$ is not a polynomial endofunctor: for example, the diagonal and antidiagonal subsets of $\{0,1\} \times \{0,1\}$ have the same projections, showing that $\mathcal{P}^{+}_{\mathrm{fin}}$ does not preserve the pullback of the two maps $\{0,1\} \to 1$. Allowing such a target adds *extensional* nondeterminism directly, although the polynomial monad List^{+} also supports the must/may separation (§9). Hence it is *target-parametricity*, rather than non-polynomiality by itself, that is our principal expressivity claim. Lists provide a useful intensional comparison.

The *source* remains the free monad on a polynomial interface with dependent certificates. Our calculation uses only ordinary categories of sets and arrows; it does not assume a fully developed dependent Sweedler double category. The must/may predicate distinction is standard in fibrational logic; in particular Aguirre and Katsumata [3] analyse finite powerset liftings in weakest-precondition semantics. We cite both the original treatment and the expanded journal account by Aguirre, Katsumata and Kura [3, 4]; their Cartesian-lifting results concern compositional weakest-precondition predicate transformers. Our proof-relevant targets will deliberately fail cartesian reindexing (Proposition 4.1), while remaining monads and supporting compositional runner restriction.

2 The certified source theory, held fixed

Let $p = (A, B)$ be a polynomial interface,

$$P(X) = \coprod_{a \in A} X^{B(a)}.$$

A specification supplies precondition witnesses $C(a)$ and guarantee witnesses $D(a, c, b)$, where $a \in A$, $c \in C(a)$, and $b \in B(a)$. For a state/certificate arrow $u : Y \to X$, write $Y_x = u^{-1}\{x\}$. Aberlé's arrow-category observation [1, §5.1] gives the certified polynomial

$$\widehat{P}(u) = \left(\coprod_{a,c} \coprod_{k:B(a)\to X} \prod_{b\in B(a)} \prod_{d\in D(a,c,b)} Y_{k(b)} \longrightarrow \coprod_{a\in A} X^{B(a)} \right). \tag{2}$$

The upper summand over (a, c) is represented by its arity

$$J_{a,c} : \coprod_{b\in B(a)} D(a, c, b) \longrightarrow B(a), \tag{3}$$

and the lower summand over a by $J^0_a = (\varnothing \to B(a))$. There is a canonical arity arrow $J^0_a \to J_{a,c}$. In particular,

$$\operatorname{dom} \widehat{P}(u) \cong \coprod_{a,c} \mathbf{Set}^{\to}(J_{a,c}, u), \qquad \operatorname{cod} \widehat{P}(u) \cong \coprod_{a} \mathbf{Set}^{\to}(J^0_a, u). \tag{4}$$

Let $\widehat{T} = \widehat{P}^*$ be the free certified monad. The codomain projection takes it to $T = P^*$; its upper fibres are Aberlé's inductively defined FreeDep programs. Thus, for any monad R on $\mathbf{Set}^{\to}$ and state arrow $e : E \to S$,

$$\operatorname{Mnd}(\widehat{T}, \operatorname{St}_e(R)) \cong \operatorname{Nat}(\widehat{P}, \operatorname{St}_e(R)), \qquad \operatorname{St}_e(R)(u) = \big(R(e \times u)\big)^e. \tag{5}$$

State passing is the conjugate monad for $e \times - \dashv (-)^e$; no additional strength hypothesis is needed. The task is therefore to choose a meaningful certified target R and calculate the right side.

3 Two proof-relevant monad liftings

Here and below $e : E \to S$ is an arbitrary arrow, neither necessarily injective nor surjective. Its elements are certified states and e forgets a certificate.

Definition 3.1 (Must and may targets)**.** Define the following arrows over $\mathcal{N}(S)$:

$$\widehat{\mathcal{N}}_{\mathsf{must}}(e) = \big(\mathcal{N}(E) \xrightarrow{\mathcal{N}(e)} \mathcal{N}(S)\big), \tag{6}$$

$$\widehat{\mathcal{N}}_{\mathsf{may}}(e) = \big(\operatorname{Sel}(e) \longrightarrow \mathcal{N}(S)\big), \tag{7}$$

$$\operatorname{Sel}(e) = \big\{(U, z) : U \in \mathcal{N}(S),\ z \in E,\ e(z) \in U\big\}. \tag{8}$$

The second map sends (U, z) to U. The first retains a nonempty finite set of certified outcomes; the second retains one selected certified member of the underlying nonempty finite set.

Theorem 3.2 (The two liftings are monads)**.** *Both endofunctors in Definition 3.1 carry monad structures and satisfy*

$$\operatorname{cod} \widehat{\mathcal{N}}_{\mathsf{must}} = \operatorname{cod} \widehat{\mathcal{N}}_{\mathsf{may}} = \mathcal{N} \operatorname{cod}$$

strictly, up to the conventional choice of product associations. Their units and multiplications project to the singleton and union operations of $\mathcal{N}$*.*

Proof. For $\widehat{\mathcal{N}}_{\mathsf{must}}$ use singleton and union at both ends; all squares commute by naturality of $\mathcal{N}$. For $\widehat{\mathcal{N}}_{\mathsf{may}}$, a commuting square $e \to e'$ with upper map $h : E \to E'$ and lower map $f : S \to S'$ takes (U, z) to $(\mathcal{N}(f)(U), h(z))$. This is well-defined because $e'h(z) = f(ez) \in \mathcal{N}(f)(U)$.

The unit at e has lower component $s \mapsto \{s\}$ and upper component

$$z \longmapsto (\{e(z)\}, z).$$

An upper element of $\widehat{\mathcal{N}}^2_{\mathsf{may}}(e)$ can be written

$$(\mathcal{U}, (U, z)), \qquad \mathcal{U} \in \mathcal{N}(\mathcal{N}S), \quad U \in \mathcal{U}, \quad e(z) \in U.$$

Define its multiplication by

$$(\mathcal{U}, (U, z)) \longmapsto \left(\bigcup \mathcal{U}, z\right). \tag{9}$$

Its lower component is union. The left unit law chooses the unique selected singleton; the right unit law sends a selected outcome into the singleton collection of its underlying set. In a triple application, both associative composites flatten all finite sets to the same union and keep the same innermost selected witness z. Hence the monad axioms hold. Naturality follows from image preservation of membership. □

Remark 3.3 (Proof-relevant, not simply predicate, liftings). If $e : E \hookrightarrow S$ is a subobject, $\widehat{\mathcal{N}}_{\mathsf{must}}(e)$ is again mono and classifies those U with $U \subseteq E$. But $\widehat{\mathcal{N}}_{\mathsf{may}}(e)$ is generally *not* mono: if a set U has two certified elements, it has two distinct selected upper witnesses. Taking the *image* of its upper arrow recovers the usual may predicate $U \cap E \neq \varnothing$. Thus the familiar Boolean modalities arise by proof-irrelevant collapse; the arrow-category may lifting retains the successful choice. We do not claim either construction is the unique fibrational lifting or that every predicate-level property automatically lifts canonically.

Remark 3.4 (No canonical comparison of certificates). A must certificate has a nonempty finite set of evidence-bearing outcomes and therefore *implies the existence* of a may certificate. It does not, in general, produce one *naturally*: selecting an element of every nonempty finite set cannot be done naturally. Indeed, evaluating a putative natural arrow map at $e : E \to 1$ would give a natural selector $\mathcal{N}(E) \to E$: its upper component takes a nonempty finite subset of E to an element of that subset. The transposition of the two elements of $E = \{0, 1\}$ fixes the subset E but fixes neither element, contradicting naturality. Accordingly, we do not posit a monad morphism $\widehat{\mathcal{N}}_{\mathsf{must}} \to \widehat{\mathcal{N}}_{\mathsf{may}}$. The modalities will be compared by their supported states, not by a fictitious canonical choice operator.

4 Reindexing certificates and predicates

Write $\mathrm{cod} : \mathbf{Set}^{\to} \to \mathbf{Set}$ for the codomain fibration. Its cartesian arrows are pullback squares. A lifting of $\mathcal{N}$ would be *Cartesian* if it carried cartesian arrows over a map $f : S \to S'$ to cartesian arrows over $\mathcal{N}f$.

Proposition 4.1 (Neither proof-relevant lifting is Cartesian)**.** *Neither* $\widehat{\mathcal{N}}_{\mathsf{must}}$ *nor* $\widehat{\mathcal{N}}_{\mathsf{may}}$ *preserves cartesian arrows of* $\mathrm{cod} : \mathbf{Set}^{\to} \to \mathbf{Set}$*. Nevertheless, after restricting to mono certificate arrows and taking the image of the upper arrow, the induced Boolean must and may predicates are stable under reindexing.*

Proof. Take $e' : E' \to S'$ and $f : S \to S'$, and put $e = f^*e'$, so $E = S \times_{S'} E'$. The comparison for the must lifting is

$$\mathcal{N}(S \times_{S'} E') \longrightarrow \mathcal{N}(S) \times_{\mathcal{N}(S')} \mathcal{N}(E'). \tag{10}$$

For $S = E' = \{0,1\}$ and $S' = 1$, the diagonal and antidiagonal subsets of $S \times E'$ are distinct but have the same image under (10): both projections are the whole two-element set. Thus the comparison is not invertible.

For may, the comparison over $\mathcal{N}(S)$ is

$$\mathrm{Sel}(f^*e') \longrightarrow \mathcal{N}(S) \times_{\mathcal{N}(S')} \mathrm{Sel}(e'), \quad (U,(s,z')) \longmapsto (U,(f[U],z')). \tag{11}$$

Take $S = \{0,1\}$, $S' = E' = 1$, f the unique map and $e' = \mathrm{id}_1$. The two certified selections over $U = \{0,1\}$—one selecting $s = 0$, the other $s = 1$—have the same image under (11). Hence may also fails to preserve cartesian arrows.

For a mono $P' \hookrightarrow S'$, the Boolean upper images are $\{U : U \subseteq P'\}$ and $\{U : U \cap P' \neq \varnothing\}$. For any $U \in \mathcal{N}(S)$,

$$f[U] \subseteq P' \iff U \subseteq f^{-1}P', \qquad f[U] \cap P' \neq \varnothing \iff U \cap f^{-1}P' \neq \varnothing.$$

These equivalences prove stability at the predicate-image level. They do not make the certificate-bearing arrow squares cartesian. □

Remark 4.2 (Comparison with Cartesian weakest preconditions). Aguirre–Katsumata–Kura [4] characterize compositional weakest-precondition transformers using appropriate Cartesian liftings in a fibrational setting. The counterexamples above show that our two *specific*, proof-relevant arrow-monad liftings are not such liftings. This is not a failure of substitution or of runner composition: both remain monads, and tight theory interpretations still act by precomposition. What fails is the stronger Beck–Chevalley-style expectation that their *certificate objects* reindex simply by pullback. This matters when relating certified runners by simulations or changing their underlying state spaces.

5 Sweedler runner calculations

Write an ordinary nonempty finite branching Mealy transition as

$$\delta_a : S \longrightarrow \mathcal{N}(B(a) \times S), \qquad F_{\mathcal{N}}(S) = \prod_{a \in A} \mathcal{N}(B(a) \times S). \tag{12}$$

For a state arrow $e : E \to S$, introduce the certified one-step outcome set

$$Z_{a,c}(E) = \coprod_{b \in B(a)} (D(a,c,b) \times E),$$

with projection

$$\pi_{a,c} : Z_{a,c}(E) \longrightarrow B(a) \times S, \qquad (b,d,z') \longmapsto (b, e(z')). \tag{13}$$

The must case requires a finite *cover* of all lower outcomes; the may case requires a *selected* certified outcome.

Definition 5.1 (Certified branching transition functors)**.** For an arrow $e : E \to S$, define two arrows with the same lower set $F_{\mathcal{N}}(S)$, whose elements are one-step ordinary outcomes $u = (U_a)_{a \in A}$:

$$\mathrm{dom}\, \widehat{F}_{\mathsf{must}}(e) = \coprod_{u=(U_a)_a \in F_{\mathcal{N}}(S)} \prod_{a \in A} \prod_{c \in C(a)} \{V \in \mathcal{N}(Z_{a,c}(E)) : \mathcal{N}(\pi_{a,c})(V) = U_a\}, \tag{14}$$

$$\mathrm{dom}\, \widehat{F}_{\mathsf{may}}(e) = \coprod_{u=(U_a)_a \in F_{\mathcal{N}}(S)} \prod_{a \in A} \prod_{c \in C(a)} \{z \in Z_{a,c}(E) : \pi_{a,c}(z) \in U_a\}. \tag{15}$$

The structure arrows forget the choices of certified outcomes, leaving u. Applying a square of state arrows to ordinary outcomes and certified outcomes makes these constructions into accessible endofunctors $\widehat{F}_{\mathsf{must}}$ and $\widehat{F}_{\mathsf{may}}$ on $\mathbf{Set}^{\to}$.

Theorem 5.2 (Must/may runner–coalgebra theorem)**.** *For $\square \in \{\mathsf{must}, \mathsf{may}\}$ and every state arrow $e : E \to S$, there is a natural bijection*

$$\boxed{\mathrm{Mnd}(\widehat{T}, \mathrm{St}_e(\widehat{\mathcal{N}}_\square)) \;\cong\; \mathbf{Set}^{\to}(e, \widehat{F}_\square(e)).} \tag{16}$$

Consequently there are isomorphisms of categories, over $\mathbf{Set}^{\to}$,

$$\mathrm{Run}(\widehat{T}, \widehat{\mathcal{N}}_\square) \cong \mathrm{Coalg}(\widehat{F}_\square). \tag{17}$$

Their data are an ordinary transition δ as in (12), together with the following upper transition for every $z \in E$, $a \in A$, $c \in C(a)$:

$$\mathsf{must}: \quad V_{a,c}(z) \in \mathcal{N}(Z_{a,c}(E)), \qquad \mathcal{N}(\pi_{a,c})(V_{a,c}(z)) = \delta_a(ez), \tag{18}$$

$$\mathsf{may}: \quad z'_{a,c}(z) \in Z_{a,c}(E), \qquad \pi_{a,c}(z'_{a,c}(z)) \in \delta_a(ez). \tag{19}$$

In both cases, the entire *lower transition δ_a is independent of the choice of c.*

Proof. The free-monad adjunction gives the first isomorphism in (5). By (4), a natural transformation $\widehat{P} \to \mathrm{St}_e(\widehat{\mathcal{N}}_\square)$ is determined by a lower element of

$$\mathrm{cod}\,\mathrm{St}_e(\widehat{\mathcal{N}}_\square)(J^0_a) \cong \mathbf{Set}(S, \mathcal{N}(S \times B(a)))$$

for each a, and an upper element of

$$\mathrm{dom}\,\mathrm{St}_e(\widehat{\mathcal{N}}_\square)(J_{a,c}) \cong \mathbf{Set}^{\to}(e, \widehat{\mathcal{N}}_\square(e \times J_{a,c}))$$

for each (a, c), with compatibility along $J^0_a \to J_{a,c}$. The lower datum is δ_a (interchanging factors).

For the must target, $\widehat{\mathcal{N}}_{\mathsf{must}}(e \times J_{a,c})$ has upper set $\mathcal{N}(E \times \coprod_b D(a,c,b))$ and lower set $\mathcal{N}(S \times B(a))$. A square from e into this arrow consists of a lower function $S \to \mathcal{N}(S \times B(a))$ and an upper function $E \to \mathcal{N}(E \times \coprod_b D)$ whose projection is the lower function applied at $e(z)$. This is precisely (18).

For the may target, its upper set consists of an underlying $U \in \mathcal{N}(S \times B(a))$ and one certified member $(z', (b, d))$ lying over a member of U. A square from e picks such a member for each certified input z, and yields (19).

Finally, the arity arrow $J^0_a \to J_{a,c}$ is the identity on lower continuation positions $B(a)$. Because $\mathrm{cod}\,\widehat{\mathcal{N}}_\square = \mathcal{N}\,\mathrm{cod}$, its image under state passing acts as the identity on the lower element. Compatibility therefore forces *the same* δ_a for every c. Yoneda conversely reconstructs the natural transformation from exactly these compatible elements. Naturality in state arrows identifies morphisms preserving (18) or (19) with coalgebra morphisms. □

Proposition 5.3 (Classical coalgebraic interpretation)**.** *Suppose $D(a, c, b) = 1$ for every (a, c, b). For each (a, c) a must-certified transition is a nonempty finitely branching coalgebra structure on E for the response functor $\mathcal{N}(B(a) \times -)$, making $e : E \to S$ a coalgebra homomorphism into δ_a. Its graph is consequently a functional bisimulation (on the reached image). A may-certified transition, by contrast, is a deterministic transition on E selecting a response and successor whose image is* one of *the transitions allowed by δ_a: its graph is a forward simulation into the ordinary nondeterministic machine.*

Proof. With $D = 1$, the map $\pi_{a,c}$ in (13) is $B(a) \times e$. The must equation is exactly

$$\mathcal{N}(B(a) \times e) \circ V_{a,c} = \delta_a \circ e,$$

the defining coalgebra-homomorphism square. The may equation asserts only that the selected image belongs to the ordinary transition set. This is precisely the stated simulation condition. □

Remark 5.4 (Preconditions can steer angelic choices). The entire *ordinary* transition δ_a remains independent of the precondition witness c, but the *selected may branch* may depend on c. Thus the certified evidence can determine which permitted execution is selected. Interpreting this as a safety guarantee presupposes an angelic choice mechanism, a scheduler, or some agent able to implement the selected branch. Under uncontrolled demonic nondeterminism, a may certificate establishes existence of a good execution, not correctness of every execution.

Remark 5.5 (What the theorem does not say). The result is about *free* certified source theories and monad maps into state passing. It does not assert that an arbitrary abstract measuring comonad $\{\widehat{T}, \widehat{\mathcal{N}}_\square\}$ exists in a preselected duoidal universe. Under the standard accessibility hypotheses the concrete one-step functors $\widehat{F}_\square$ admit cofree coalgebra constructions, giving comonads with precisely the displayed coalgebras; identifying these with a universal Sweedler hom is an additional ambient-structure theorem. Nor does the argument yet impose equations on the certified source.

6 Compositionality under verified program interpretations

Suppose $\widehat{T}_p \xrightarrow{\widehat{f}} \widehat{T}_q$ is a verified tight interpretation: a monad map on the arrow category lying over an ordinary free-theory interpretation $T_p \to T_q$. Such maps are exactly the set-level encoding of Aberlé's verified polynomial-program implementations in the companion note; their upper components carry evidence along lower-level program trees. The present target is arbitrary among the two liftings, so the same argument applies to both.

Proposition 6.1 (Certified program restriction). *For either* $\square \in \{\mathsf{must}, \mathsf{may}\}$*, precomposition with* $\widehat{f}$ *induces a functor*

$$\widehat{f}^* : \mathrm{Run}(\widehat{T}_q, \widehat{\mathcal{N}}_\square) \longrightarrow \mathrm{Run}(\widehat{T}_p, \widehat{\mathcal{N}}_\square). \tag{20}$$

The functor leaves the arrow of states unchanged, agrees with ordinary nondeterministic execution after forgetting certificates, and composes strictly with further verified interpretations (up to the standard conventions for their presentation).

Proof. A runner for $\widehat{T}_q$ with states e is a monad map $\alpha : \widehat{T}_q \to \mathrm{St}_e(\widehat{\mathcal{N}}_\square)$. Send it to $\alpha \circ \widehat{f}$. Since both factors are monad maps, so is the composite. A morphism of runners is preserved because $\widehat{f}$ is independent of residual states. Associativity of precomposition gives functoriality and the composition law. Taking codomains yields the usual translation of source operations into lower-level nondeterministic computations. □

An inductive/coinductive interaction. The source monad $\widehat{T}_p$ contains inductively generated certified programs, while the state-passing action can be iterated over indefinitely evolving certified machines. The interpretation $\alpha \circ \widehat{f}$ is exactly where well-founded evidence-bearing programs act on potentially coinductive behaviour. For must, Kleisli union aggregates certified outcomes of all branches. For may, a selected certified branch is threaded through

the composite, as guaranteed by (9); using this as an actual execution strategy requires control over branch selection, as noted after Proposition 5.3. This is the same operational pattern as the deterministic `run-mealyD` construction [1], with the target now controlling nondeterministic effects.

7 Certified viability: two greatest fixed points

The distinction can be described without choosing concrete certificates. Fix an ordinary nondeterministic Mealy machine δ on S and a set $G \subseteq S$ of states allowed to be certified. A boolean support $V \subseteq G$ is viable if the inclusion $V \hookrightarrow S$ can carry a must or may certificate for the restricted machine. Because all command and guarantee families are fixed, write $D(a, c, b) \neq \varnothing$ for the existence of a corresponding postcondition witness.

Definition 7.1 (Viability operators)**.** For $V \subseteq S$ define monotone operators

$$\Phi_{\mathsf{must}}(V) = G \cap \{s \in S : \forall a \in A,\ \forall c \in C(a),\ \forall (b, s') \in \delta_a(s), \tag{21}$$
$$D(a, c, b) \neq \varnothing \text{ and } s' \in V\}, \tag{22}$$
$$\Phi_{\mathsf{may}}(V) = G \cap \{s \in S : \forall a \in A,\ \forall c \in C(a),\ \exists (b, s') \in \delta_a(s), \tag{23}$$
$$D(a, c, b) \neq \varnothing \text{ and } s' \in V\}. \tag{24}$$

The quantification over c is vacuous if $C(a) = \varnothing$; such a command has no certification obligation, though its ordinary transition still exists.

Theorem 7.2 (Supported states are viability kernels)**.** *The largest subset of G that admits a proof-irrelevant must-certified structure over the fixed machine δ is $\nu\Phi_{\mathsf{must}}$; the analogous may support is $\nu\Phi_{\mathsf{may}}$. More generally, if an arbitrary certified state arrow $e : E \to S$ has image contained in G, then $\mathrm{im}(e) \subseteq \nu\Phi_{\square}$ for the corresponding modality. Consequently*

$$\nu\Phi_{\mathsf{must}} \subseteq \nu\Phi_{\mathsf{may}}. \tag{25}$$

If each $C(a)$ and $D(a, c, b)$ is a singleton, these reduce to the familiar operators

$$G \cap \{s : \forall a,\ \delta_a(s) \subseteq B(a) \times V\}, \qquad G \cap \{s : \forall a,\ \delta_a(s) \cap (B(a) \times V) \neq \varnothing\}.$$

Proof. Let $V \subseteq G$ support a certified runner with the inclusion $V \hookrightarrow S$ as its certificate arrow. For each $s \in V$, the must law (18) implies that for every command and precondition witness, each ordinary outcome has a guarantee witness and a successor in V. Thus $V \subseteq \Phi_{\mathsf{must}}(V)$. The may law similarly gives $V \subseteq \Phi_{\mathsf{may}}(V)$. The same reasoning with $V = \mathrm{im}(e)$ works for a non-monic state arrow, since every certified successor has underlying state in the image. By Knaster–Tarski any such V is contained in the corresponding greatest fixed point.

Conversely, on a post-fixed point V, for each $s \in V, a, c$ choose guarantee witnesses for eligible ordinary outcomes. In the must case take one certified representative for each member of the *finite nonempty* set $\delta_a(s)$; its collection is a finite nonempty upper set projecting onto all of $\delta_a(s)$. In the may case select one eligible outcome and a witness. Classical choice across the (possibly infinite) family of states and commands makes these selections coherent as functions. They give runners on $V \hookrightarrow S$ by Theorem 5.2. The union of all post-fixed subsets is $\nu\Phi$, so the claimed supports follow. The inclusion (25) holds because any nonempty finite set all of whose outcomes are eligible has at least one eligible outcome. □

Remark 7.3 (A qualification about witnesses). Taking $G = S$ gives the unconstrained kernels. The support theorem uses classical set theory and choice to select witnesses. It gives the *largest support* admitting some certificates, not a canonical family of certificates or an equivalence of their categories. Moreover it concerns a *fixed* underlying δ and a free source; equations, other targets, and dependence of C on the state may change the formula.

8 Two worked machines

8.1 A safe branch and an unsafe branch

Take a unary command tick with response set 1, trivial precondition and guarantee witnesses, and states $S = \{g, b\}$. Let

$$\delta(g) = \{g, b\}, \qquad \delta(b) = \{b\}, \qquad G = \{g\}.$$

In a picture, g can stay safe or enter b, and b loops forever. The certificate arrow $e : \{g\} \hookrightarrow S$ supports a may runner: when the current certified state is g, select the g-branch. It does *not* support a must runner: no finite collection of certified outcomes can project onto $\{g, b\}$. Theorem 7.2 calculates

$$\boxed{\nu\Phi_{\mathsf{may}} = \{g\}, \qquad \nu\Phi_{\mathsf{must}} = \varnothing.}$$

This witnesses a strict difference between the two targets with *identical source theory and underlying operational machine.*

8.2 Proof relevance: staying or doubling a sum of squares

To exhibit more than Boolean certification, let the ordinary state space be $S = \mathbb{N}$ and take

$$Q(n) = \{(x, y) \in \mathbb{Z}^2 : x^2 + y^2 = n\}, \qquad e : \sum_{n \in \mathbb{N}} Q(n) \longrightarrow \mathbb{N}. \tag{26}$$

A state such as 25 has multiple witnesses, e.g. $(3, 4)$ and $(0, 5)$, and we do not identify them. Consider one unary command with trivial input and output evidence, interpreted nondeterministically by

$$\delta(n) = \{n, 2n\}.$$

Starting from the certified state $z = (n, x, y)$, two certified successors are available:

$$z_{\text{stay}} = (n, x, y), \qquad z_{\text{double}} = (2n, x - y, x + y).$$

Indeed $(x-y)^2 + (x+y)^2 = 2(x^2+y^2) = 2n$. A *must* runner assigns the finite upper outcome set

$$V(z) = \{z_{\text{stay}}, z_{\text{double}}\}, \qquad \mathcal{N}(e)(V(z)) = \{n, 2n\} = \delta(ez). \tag{27}$$

A *may* runner can select either certified successor. For $n = 25$ with witness $(3, 4)$, the doubling branch yields $(50, -1, 7)$; with witness $(0, 5)$ it yields $(50, -5, 5)$. These are distinct *runner states* above the same ordinary doubled state, but in this unary theory with $C = D = 1$ they do not define distinct *final must-certified behaviours*: the guarantee exposes no certificate data. Indeed, with singleton C and D, if $\beta : B \to F_{\mathcal{N}}(B)$ is the final ordinary coalgebra, then $\mathrm{id}_B : B \to B$, with upper and lower transition β, is the final must-certified coalgebra. Every certified $e : E \to S$ maps to it uniquely: the lower map is $\mathrm{beh} : S \to B$ and the upper map is $\mathrm{beh}\ e$. Thus each final must-fibre has precisely one element. In the may case, the final fibres instead record selected branch behaviour, not the arbitrary witnesses stored in E. At $n = 0$ the set notation automatically removes duplicates, as required by the powerset semantics.

To make the witness observable, add a command show with response set $B(\mathsf{show}) = \mathbb{N}$, precondition $C(\mathsf{show}) = 1$, and guarantee

$$D(\mathsf{show}, *, b) = Q(b).$$

Its ordinary transition at state n is the singleton $\delta_{\mathsf{show}}(n) = \{(n, n)\}$. On certified state $z = (n, x, y)$, both modalities can return the certified outcome

$$(b, d, z') = (n, (x, y), (n, x, y)).$$

In particular, at $n = 25$, the witnesses $(3, 4)$ and $(0, 5)$ yield different *observable guarantee evidence* on show. They therefore determine different behaviours in the final certified object for the enriched interface. The source now genuinely observes proof relevance rather than merely storing extra distinguished states. A nontrivial precondition such as the multiplicative sum-of-squares example in the companion note can also be combined with these liftings.

9 What changes with the nonempty-list monad?

The nonempty-list monad

$$\mathrm{List}^+ X = \coprod_{n \geq 1} X^n$$

is polynomial on **Set**, whereas $\mathcal{P}^+_{\mathrm{fin}}$ is not. Both admit versions of the two arrow liftings:

$$\widehat{L}_{\mathsf{must}}(e) = (\mathrm{List}^+ E \longrightarrow \mathrm{List}^+ S),$$
$$\widehat{L}_{\mathsf{may}}(e) = (\{(\ell, i, z) : 1 \leq i \leq |\ell|,\ e(z) = \ell_i\} \longrightarrow \mathrm{List}^+ S).$$

In the may lifting, an upper witness selects a *position* in the underlying list, not merely a value. Multiplication concatenates lists and transports the selected occurrence to its corresponding position in the concatenation; this defines the monad laws in the same way that (9) does for sets.

There is a monad morphism

$$q : \mathrm{List}^+ \longrightarrow \mathcal{P}^+_{\mathrm{fin}}, \qquad [x_1, \ldots, x_n] \longmapsto \{x_1, \ldots, x_n\}, \tag{28}$$

which discards order and repetitions. It lifts to both must and may constructions; in the may case (ℓ, i, z) maps to $(\operatorname{supp} \ell, z)$. These commute with units and multiplications. Thus the extensional theories are quotients of the intensional list theories at the target.

For example, lists $[g, g]$ and $[g]$ describe different list computations but the same powerset outcome $\{g\}$. A may-list certificate can distinguish selection of the first or second occurrence of g in $[g, g]$, even if both carry the same proof witness. If ordering, scheduling or branch multiplicity is semantically relevant, this is useful. If the intended claim is *may succeed* versus *all nondeterministic outcomes safe*, those distinctions are distractions. Hence $\mathcal{P}^+_{\mathrm{fin}}$ is the preferable main target here, with List^+ as a polynomial comparison rather than the default semantics.

10 Allowing deadlock: partial correctness versus possible success

So far $\mathcal{N} = \mathcal{P}^+_{\mathrm{fin}}$ excludes the empty outcome. The ordinary finite-powerset monad $\mathcal{P}_{\mathrm{fin}}$ includes it and admits the *same two formulas* for arrow liftings, with the finite sets U now possibly empty. The must upper set V may also be empty; the may upper part $\{(U, z) : e(z) \in U\}$ contains no witness over $U = \varnothing$. The monad units and multiplication remain valid, by the proofs in §3.

Proposition 10.1 (Opposite responses to deadlock)**.** *For a command with at least one precondition witness, suppose $\delta_a(s) = \varnothing$. The must equation allows a certified state above s, using the empty certified outcome set $V = \varnothing$. The may equation admits no certified state above s, since there is no member to select.*

Proof. For must, $\mathcal{P}_{\mathrm{fin}}(\pi)(\varnothing) = \varnothing = \delta_a(s)$. For may, the existential condition $\exists z' \in Z_{a,c}(E) : \pi(z') \in \varnothing$ is false. □

Thus deadlock makes must-certification vacuous on failure, in the familiar *partial-correctness* reading, while may-certification requires at least one continuing or successful outcome. Additional modelling choices are needed if deadlock is to count as an observable error rather than vacuous termination. This is another instance in which the target monad controls the verification modality.

11 Scope and next questions

The preceding results isolate a concrete expressivity gain relative to deterministic certified Mealy semantics. The source polynomial, its FreeDep monad, and the state-passing definition never change; choosing a target lifting distinguishes two forms of verified branching. In particular, a target choice contributes more than an implementation detail: it supplies an *effect-sensitive certification modality*. It would be too strong to identify a complete program logic with only the target lifting, since the source still supplies preconditions, postconditions and equations.

There are four limitations worth keeping visible. First, the present source is *free*; quotient theories with certified equations may impose additional stability laws. Second, the empty-outcome extension in §10 produces a partial-correctness/possible-success distinction, but the interpretation of deadlock as failure, termination, or error remains a modelling choice. Third, this is a set-level theorem: proof types with higher identifications are not represented. Fourth, the concrete coalgebra functors are calculated without first constructing a closed duoidal equipment for every monad on $\mathbf{Set}^{\to}$. These are suitable next problems, not hypotheses secretly used above. A particularly important next question is how the non-Cartesian witness comparisons (10)–(11) interact with runner simulations and 2-cells: the ordinary predicate images commute with reindexing, but explicit certificates do not.

An ambient-independent theorem should explain when the same source and target liftings exist over a suitable category of modes and certificates, and when their measuring comonads are stable under change of ambient. Meanwhile the example already shows the intended principle in a genuinely external setting: Aberlé's source verification can be paired with two distinct nondeterministic residual semantics, and the resulting certified machines are recovered by exactly the same Sweedler runner operation.

References

[1] C. B. Aberlé, *Compositional Program Verification with Polynomial Functors in Dependent Type Theory*, arXiv:2604.01303 (2026). https://arxiv.org/abs/2604.01303.

[2] D. McDermott, E. Rivas and T. Uustalu, *Sweedler Theory of Monads*, FoSSaCS 2022, LNCS 13242, pp. 428–448. https://doi.org/10.1007/978-3-030-99253-8_22.

[3] A. Aguirre and S.-y. Katsumata, *Weakest Preconditions in Fibrations*, Electronic Notes in Theoretical Computer Science **352** (2020), 5–27. https://doi.org/10.1016/j.entcs.2020.09.002.

[4] A. Aguirre, S.-y. Katsumata, and S. Kura, *Weakest preconditions in fibrations*, Mathematical Structures in Computer Science **32**(4) (2022), 472–510. https://doi.org/10.1017/S0960129522000330.

[5] S. Libkind and D. I. Spivak, *Pattern runs on matter: The free monad monad as a module over the cofree comonad comonad*, Applied Category Theory 2024; arXiv:2404.16321. https://arxiv.org/abs/2404.16321.