

Getting to Grips with Sweedler Runners for Systems

A conversational tutorial, Part VII: higher-dimensional state
Precubical sets, true concurrency, and the boundary between cartesian and cubical effects

October 2026

Aim. Parts I–III developed runners for ordinary Set-based algebraic theories; Parts IV–VI began to vary the ambient world in which those theories act. Part VII makes a more radical change: the state object itself becomes an *execution geometry*. A precubical set has vertices, edges, squares, cubes, and higher-dimensional cells, so a state object can remember not merely where a computation is but how independent activities overlap.

There is, however, a new distinction which will organize the whole chapter. Merely putting an ordinary Set-based theory in a cubical presheaf category does *not* make the effect genuinely cubical. Such theories lift pointwise: they act on every dimension, but they do not themselves create, consume, or reorganize cubical dimension. To obtain effects which genuinely interact with concurrency, the theory must itself use the dimension-adding cubical tensor. Thus the main question is not simply “can runners have cubical state?” but:

when does an effect merely travel along cubical geometry, and when does it use it?

The answer will lead us from ordinary Lawvere theories toward theories enriched or linear in the cubical tensor, and hence toward the PROP-like structures that will become increasingly important for Petri nets and other concurrent systems.

Contents

1	Why ordinary transition systems lose concurrency	3
2	Precubical sets: cubes without degeneracies	3
3	From a precubical set to an HDA	4
4	The one-dimensional shadow, and what filling adds	5
5	Equations and cells live at different corners	5
6	The dimension-adding tensor	6
6.1	What changes after symmetrising	6
7	Why ordinary precubical sets are not enough for the symmetric Sweedler base	6
8	The duoidal pair is non-normal	7

9	Two different ways theories can live in the cubical world	8
9.1	Cartesian theories and cartesian runners	8
9.2	Cubical-linear theories and cubical runners	8
10	What the conservative Set-based lift actually gives	8
10.1	Observations are constant along execution components	9
10.2	The Store warning	9
11	The dimension-zero obstruction for cubical comonoids	9
11.1	Reader becomes dimension-zero	10
11.2	Store and discarding	10
12	Three embeddings of sets, and why “discrete” is ambiguous	10
13	Revisiting Moore machines	11
14	Levelwise monads: what lifts automatically	11
15	Is an HDA itself a comodel? Two answers	12
15.1	The unary presheaf presentation	12
15.2	The stronger enriched question	12
16	Probability and nondeterminism on cubical executions	13
17	The candidate Sweedler ambient category	13
18	Petri nets: why the linear side now looks inevitable	14
19	Behavioural equivalence and recent HDA formalisms	15
20	Toward a Sweedler calculus of concurrency	15
20.1	Two processes and two locks	15
20.2	A compositional summary: the lock-order relation	16
20.3	Repairing the example	17
20.4	Where the Sweedler structure should enter	18
20.5	From deadlock freedom to a progress calculus	19
20.6	A first research target	20
21	What Part VII has established, and what remains open	20
22	The larger lesson	21
	References	22

1 Why ordinary transition systems lose concurrency

Consider two actions, a and b , which may happen independently. A transition system can draw the two possible execution orders

$$s_{00} \xrightarrow{a} s_{10} \xrightarrow{b} s_{11}, \quad s_{00} \xrightarrow{b} s_{01} \xrightarrow{a} s_{11}.$$

But this picture alone does not tell us whether these paths represent two unrelated choices or two boundary traces of one concurrent execution. Interleaving semantics has replaced concurrency by a choice between linear histories.

A higher-dimensional automaton adds the missing interior. We draw

$$\begin{array}{ccc} s_{00} & \xrightarrow{a} & s_{10} \\ \downarrow b & & \downarrow b \\ s_{01} & \xrightarrow{a} & s_{11} \end{array}$$

and then declare that this boundary is filled by a 2-cell σ . The cell is additional data: it says that a and b can overlap as one two-dimensional execution. A hollow square can have exactly the same vertices and edges without containing this witness.

Interleaving versus true concurrency. A graph records states and one-dimensional transitions. A precubical set can additionally record several transitions as simultaneously active. An n -cube represents an n -dimensional piece of concurrent execution.

Higher-dimensional automata were developed precisely to retain causal and overlapping structure which interleaving models can erase; van Glabbeek's comparison of concurrency models is a useful point of entry, with the published erratum supplying the corrected figures [1, 2]. Modern HDA work continues to use cubical or closely related presheaf geometries for non-interleaving semantics [6, 9].

2 Precubical sets: cubes without degeneracies

Let $\square$ be the small cube category generated by objects

$$[0], [1], [2], \dots$$

and coface maps

$$\delta_i^\epsilon : [n-1] \longrightarrow [n], \quad \epsilon \in \{0, 1\}, \quad 1 \leq i \leq n,$$

subject to the cubical relations expressing the compatibility of inserting two fixed coordinates in either order. A *precubical set* is a presheaf

$$X : \square^{\text{op}} \longrightarrow \mathbf{Set}, \quad X_n := X([n]).$$

The cofaces induce face maps

$$d_i^\epsilon : X_n \longrightarrow X_{n-1}$$

satisfying

$$d_i^\epsilon d_j^\eta = d_{j-1}^\eta d_i^\epsilon \quad (i < j).$$

Thus

$$\mathbf{PreCub} = [\square^{\text{op}}, \mathbf{Set}].$$

The low dimensions have the intended execution reading:

Dimension	Interpretation
X_0	vertices, or ordinary states
X_1	directed edges, or one active event
X_2	squares, or two events active concurrently
X_3	cubes, or three-way concurrency
X_n	n -dimensional concurrent execution data

The representable

$$\square[n] := \square(-, [n])$$

is the standard n -cube, and every precubical set is canonically assembled from these representables by colimits.

Why “pre”cubical? Cubical sites come in several forms. One may add degeneracies, connections, symmetries, reversals, or other maps. The minimal precubical category retains just the faces needed for directed cubes. Later we will deliberately add symmetries because our Sweedler architecture wants a symmetric monoidal base.

3 From a precubical set to an HDA

Fix an alphabet Σ . A standard labelled precubical presentation uses a label object $!\Sigma$ with

$$(!\Sigma)_n = \Sigma^n,$$

and face maps deleting the appropriate coordinate. A labelled precubical set is a map

$$\lambda : X \longrightarrow !\Sigma.$$

At dimension one this labels edges; at dimension two it forces opposite sides of a square to carry the corresponding repeated event labels. A basic HDA then consists of

$$(X, \lambda, I, F),$$

where $I, F \subseteq X_0$ are designated initial and accepting states. Different HDA formalisms refine the labelling and boundary data in different ways, so we will use only what is common to the present discussion [6, 9].

An execution can still be observed along one-dimensional boundary paths. A square may have traces

$$a; b \quad \text{and} \quad b; a,$$

but the HDA remembers that these traces bound the same two-dimensional execution. More developed semantics extract interval orders, track objects, directed homotopy information, languages, and behavioural equivalences from this geometry [7, 8, 9].

An HDA remembers execution shape, not only reachable endpoints.

4 The one-dimensional shadow, and what filling adds

Let $\mathrm{tr}_1 X$ denote the 1-truncation of a precubical set. It is the directed graph

$$X_1 \rightrightarrows X_0$$

with source and target d_1^0, d_1^1 . Conversely, a directed graph G has a 1-skeleton $\mathrm{sk}_1 G$ with no nontrivial cells above dimension one. At the other extreme $\mathrm{cosk}_1 G$ fills every higher-dimensional boundary whose faces are compatible. Thus among precubical sets with a fixed graph-shaped 1-truncation there is a canonical range

$$\mathrm{sk}_1 G \longrightarrow X \longrightarrow \mathrm{cosk}_1 G.$$

The left endpoint has no higher witnesses; the right endpoint contains every compatible higher filling. This makes precise the statement that many cubical executions can have the same interleaving shadow.

The cubical tensor introduced below sharpens the picture further. If $X \otimes_{\square} Y$ is the dimension-adding tensor, then its 1-truncation depends only on the two 1-skeleta and is their graph box product. Informally,

$$\boxed{\otimes_{\square} = \text{interleaving boundary plus its canonical higher fillings.}}$$

In particular sk_1 is not monoidal for $\otimes_{\square}$: tensoring two one-dimensional automata automatically generates the square witnessing independence of one edge from the other.

This is different from the pointwise cartesian product. For example, $\square[1] \times \square[1]$ pairs equal-dimensional cells and has a lockstep or synchronous flavour, whereas $\square[1] \otimes_{\square} \square[1]$ is genuinely two-dimensional. We will soon see that the pair

$$(\otimes_{\square}, \times)$$

is therefore not a technical redundancy: it separates asynchronous composition from levelwise pairing.

5 Equations and cells live at different corners

Part III introduced tensor products of theories. If T_a and T_b each contain a unary operation, a theory-level equation can impose

$$a(b(x)) = b(a(x)).$$

An HDA square looks similar in a picture, but it lives at a different level. The equation is a *uniform law of the theory*; a filled square is a *particular witness in one state/execution object*. Comparing them directly therefore mixes two corners of the architecture.

A better four-corner comparison is:

	Algebraic/theory side	Geometric/state side
Uniform structure	an equation such as $ab = ba$	a natural family of coherent squares
Particular behaviour	an instance of the equation	one specified filled square $\sigma \in S_2$

An ordinary equation can say that two composites agree, but it cannot remember a non-uniform witness attached to one execution. Conversely, if a theory itself had cubically enriched hom-objects, then a 2-cell in the theory could provide a uniform family of square witnesses. This observation will return when we consider enriched Lawvere theories.

6 The dimension-adding tensor

The presheaf category **PreCub** has a cartesian product

$$(X \times Y)_n = X_n \times Y_n.$$

The cube category also has concatenation of coordinates,

$$[p] \otimes [q] = [p + q],$$

and Day convolution along this monoidal structure gives a second product

$$X \otimes_{\square} Y.$$

On representables,

$$\square[p] \otimes_{\square} \square[q] \cong \square[p + q],$$

so in particular

$$\boxed{\square[1] \otimes_{\square} \square[1] \cong \square[2].}$$

For ordinary ordered precubical sets the cells may be described by

$$(X \otimes_{\square} Y)_n \cong \coprod_{p+q=n} X_p \times Y_q.$$

This is the cleanest formula for geometric intuition: dimensions add.

6.1 What changes after symmetrising

The preceding coproduct formula is *not* the formula in the symmetric cube category. Once coordinate permutations are admitted, each split $n = p + q$ comes with the possible shuffles of the two coordinate blocks. For symmetric precubical sets one obtains schematically

$$(X \otimes_{\square} Y)_n \cong \coprod_{p+q=n} \text{Sh}(p, q) \times X_p \times Y_q,$$

with the symmetric-group actions identifying the equivalent descriptions inside each block. Thus $\square_s[1] \otimes_{\square} \square_s[1]$ has two 2-cells corresponding to the two shuffles, just as $\square_s[2]$ carries its S_2 -action. The passage from ordered to symmetric cubes must therefore be made explicitly, for example by the free-symmetric/left-Kan-extension construction rather than by reusing ordered formulas verbatim [4, 5].

Symmetric cubes also make room for genuine automorphisms of a cell. In particular, a 2-cell may be fixed by the coordinate transposition; when this occurs its corresponding faces are identified in the symmetric way. This is one route toward representing autoconcurrency, though the exact operational reading depends on the chosen HDA convention.

7 Why ordinary precubical sets are not enough for the symmetric Sweedler base

The minimal precubical tensor is monoidal but not braided. A short obstruction already appears on representables. Any hypothetical braiding would give an automorphism

$$\square[1] \otimes_{\square} \square[1] \cong \square[2].$$

But the ordered 2-cube has no coordinate-swapping automorphism in the minimal cube category. Naturality with respect to a coface $\delta : [0] \rightarrow [1]$ would then force

$$\delta \otimes \text{id} = \text{id} \otimes \delta : [1] \longrightarrow [2],$$

although the two maps free different coordinates. Hence no such braiding exists.

For the symmetric Sweedler architecture developed in the previous parts, this removes the tempting shortcut “just work braidedly in ordinary **PreCub**.” One could instead develop a genuinely nonsymmetric version of the measuring theory, but that would be a different project. Here we take the direct route and use a symmetric cubical site. Following the tradition of extended/symmetric cubical sites [4, 5], write

$$\mathcal{E}_{\square} = \mathbf{sPreCub}.$$

Its Day tensor $\otimes_{\square}$ is symmetric and dimension-adding.

There is also now a third option in the HDA literature which is worth keeping visible. Recent work organizes cells by *concurrency lists* (conclists), finite ordered lists of currently active events, and presents HDAs as presheaves over the resulting category [9]. The event order itself keeps track of coordinates, including repeated labels. This may eventually be a more natural systems base than symmetric precubical sets. We will not identify the two frameworks here; for the present chapter **sPreCub** remains the working symmetric base, while conclists are an important comparison point.

8 The duoidal pair is non-normal

Any monoidal structure on a cartesian category interacts canonically with the cartesian product. In our case there is an interchange map

$$\zeta : (A \times B) \otimes_{\square} (C \times D) \longrightarrow (A \otimes_{\square} C) \times (B \otimes_{\square} D),$$

obtained from the tensor of the relevant projections. Thus the cubical and cartesian products form a canonical duoidal pair of the orientation needed to compare the two kinds of structure.

But this duoidal structure is *not normal*. The unit for the cubical tensor is the representable point

$$\square[0],$$

whereas the cartesian unit is the terminal constant presheaf

$$\Delta 1.$$

They are different objects. This distinction will repeatedly surface below. In particular, the innocent phrase “regard a set as a discrete cubical object” conceals several inequivalent embeddings whose difference is exactly visible at the two units.

Semantically one may read

$\otimes_{\square}$ as asynchronous, dimension-adding composition, $\times$ as levelwise or lockstep pairing.

The interchange ζ then says that independent composition can be projected coherently into two synchronous components. We will use this reading only as intuition; the formal point is simply that the two tensors must not be silently interchanged.

9 Two different ways theories can live in the cubical world

We can now state the structural issue that governs the rest of the chapter. Suppose T is an ordinary finitary Set-based theory. Its arities are cartesian: an n -ary operation is interpreted using cartesian powers or, dually for comodels, coproducts of n copies of the state. By contrast, the runner form

$$\theta_X : TX \otimes_{\square} S \longrightarrow R(X \otimes_{\square} S)$$

uses the dimension-adding cubical tensor. These are different monoidal structures.

For the ordinary Set-based runner/comodel correspondence, one evaluates naturality at a generic object encoding the arity. In the cubical base this cannot simply be repeated with $\otimes_{\square}$. If k is a finite set and Δk the constant presheaf, then in general

$$\Delta k \otimes_{\square} S \not\cong k \cdot S.$$

The left side adds cubical dimensions coming from the constant object's positive-dimensional cells, whereas the right side is the ordinary copower required by a cartesian Lawvere theory.

There are therefore two clean constructions.

9.1 Cartesian theories and cartesian runners

An ordinary Lawvere theory can be interpreted internally using $\times$ and ordinary copowers. Its comodels in **sPreCub** are perfectly meaningful. This is the conservative lift of the Set-based theory.

9.2 Cubical-linear theories and cubical runners

Alternatively, one may build a theory whose arities are native to $\otimes_{\square}$, for example using finitely presentable cubical arities A and the internal homs associated to the closed structure. Runners for such a theory naturally have the form

$$TX \otimes_{\square} S \longrightarrow R(X \otimes_{\square} S).$$

This is the setting in which the effect itself can create, consume, or redistribute execution dimension.

A mixed calculus should be possible using the duoidal interchange ζ , but that interaction must be constructed explicitly. It cannot be obtained by silently replacing $\times$ by $\otimes_{\square}$.

The thesis of Part VII. Set-derived cartesian effects can be lifted to cubical state, but the lift is pointwise: the effect travels along the cubical geometry rather than generating it. Genuine higher-dimensional effects require theory arities which are themselves cubical-linear, enriched, or PROP-like with respect to $\otimes_{\square}$.

10 What the conservative Set-based lift actually gives

Let T be an ordinary Lawvere theory. A T -comodel needs coproducts, and coproducts in a presheaf category are computed pointwise. Hence there is a canonical equivalence

$$\boxed{\mathbf{Comod}_T(\mathbf{PreCub}) \simeq [\square^{\text{op}}, \mathbf{Comod}_T(\mathbf{Set})]}$$

and likewise in the symmetric cubical presheaf category. Thus a cubical T -comodel is simply a presheaf of ordinary T -comodels with all face maps comodel morphisms.

This result is useful, but it is much more conservative than saying that the effect “preserves genuine concurrency.” The effect acts at every dimension and is natural with respect to faces, yet it does not itself produce or consume a cell. The cubical geometry is carried along as an indexing geometry.

10.1 Observations are constant along execution components

Suppose the theory contains an observation with result set O . Its comodel cooperation has type

$$o : S \longrightarrow O \cdot S = \coprod_{o \in O} S.$$

A map of presheaves into this coproduct must choose one summand compatibly with every face map. Hence the chosen observation is constant along every connected piece of execution geometry: the endpoints of an edge, all vertices of a filled square, and more generally all cells connected through faces carry the same observed value.

This is not a defect in the theorem; it tells us exactly how much information an ordinary Set-derived observation can extract from a cubical state. If we want an observation that varies as an execution proceeds, then the observation object or the theory itself must carry richer cubical structure.

10.2 The Store warning

The same phenomenon appears for Store. In the pointwise lift, a stored Set-value is part of the ordinary comodel structure in every degree, and the face maps must preserve it. Thus the stored value is constant along a cubical execution cell. A genuinely cubical store whose contents change along faces is a different effect, not the naive internalization of the ordinary Set-theory.

11 The dimension-zero obstruction for cubical comonoids

There is a second diagnostic which explains why familiar cartesian effects become restrictive when forced directly into $\otimes_{\square}$. Suppose R is a comonoid for the cubical tensor. Its counit has the form

$$\varepsilon : R \longrightarrow \square[0].$$

But $\square[0]$ has no cells in positive dimensions. Since maps of precubical sets preserve dimension, R_n must be empty for every $n > 0$. Thus R is concentrated in dimension zero, and counitality forces the comultiplication to be the ordinary diagonal there. Consequently, for the cubical tensor,

$$\boxed{\mathbf{Comon}(\mathcal{E}_{\square}, \otimes_{\square}) \simeq \mathbf{Set}}$$

via the 0-skeleton embedding; in the symmetric setting the same statement applies to cocommutative comonoids.

This is a cubical analogue of the familiar Fox-style phenomenon that cartesian copying and deleting picks out a small sector inside a genuinely noncartesian tensor.

11.1 Reader becomes dimension-zero

A reader construction of the form $[R, -]$ with respect to the cubical closed structure becomes monadic in the familiar way only when the parameter R supplies the appropriate comonoid structure. The preceding proposition therefore forces such an R into dimension zero. A native cubical reader cannot simply “read a higher-dimensional object” while retaining the ordinary reader equations; some different linear or graded notion is needed.

11.2 Store and discarding

Likewise, presentations of Store which use projection or discarding maps should not be assumed to survive unchanged under $\otimes_{\square}$. If a cubical store parameter has positive-dimensional cells, laws whose interpretation requires deleting one tensor factor face the same obstruction. The precise statement depends on the chosen presentation of Store, so we record this as a warning rather than a general impossibility theorem.

The point is not that Set-derived effects literally ignore every positive-dimensional cell: their pointwise lifts act in every degree. The stronger obstruction appears when one asks the *effect structure itself* to use cubical copying, deleting, or arities. Then the cartesian sector collapses to dimension zero.

12 Three embeddings of sets, and why “discrete” is ambiguous

The Moore example in the earlier draft hid an ambiguity. For the 0-cell evaluation functor there are adjoints

$$\mathrm{sk}_0 \dashv \mathrm{ev}_0 \dashv \mathrm{cosk}_0,$$

and independently connected components are left adjoint to the constant presheaf functor,

$$\pi_0 \dashv \Delta.$$

These embeddings behave differently.

For a set A , $\mathrm{sk}_0 A$ has only 0-cells, while ΔA has a copy of A in every degree with identity restriction maps. Consequently

$$(\mathrm{sk}_0 A \times S)_n = \emptyset \quad (n > 0),$$

whereas

$$\Delta A \times S \cong A \cdot S$$

is the pointwise copower of S by A . The same copower can also be expressed using the cubical tensor as

$$A \cdot S \cong \mathrm{sk}_0 A \otimes_{\square} S.$$

This equality is useful because it shows exactly where a Set-valued arity may enter the cubical tensor without introducing spurious positive-dimensional cells.

Observations exhibit the differences even more clearly:

- a map $S \rightarrow \mathrm{sk}_0 O$ can exist only when S itself has no positive-dimensional cells;
- a map $S \rightarrow \Delta O$ is equivalently a function $\pi_0 S \rightarrow O$, so it is constant on execution components;

- a map $S \rightarrow \text{cosk}_0 O$ may encode an arbitrary function $S_0 \rightarrow O$, because higher cells of $\text{cosk}_0 O$ remember all compatible tuples of vertex values, but this is no longer the ordinary comodel cooperation $S \rightarrow O \cdot S$.

The two objects $\square[0] = \text{sk}_0 1$ and $\Delta 1$ are precisely the two units encountered above. Thus the ambiguity of “discrete” is another face of non-normality.

13 Revisiting Moore machines

Let $T_{\text{Moore}}(A, O)$ be the ordinary Set-based Moore theory from the earlier parts. Its conservative cubical lift should be read through the pointwise-comodel theorem, not as an instance of a $\otimes_{\square}$ -runner obtained by changing one symbol.

The transition cooperation has the ordinary copower form

$$\delta : S \longrightarrow A \cdot S,$$

or equivalently may be unpacked into the corresponding family of maps according to the chosen presentation. Because

$$A \cdot S \cong \Delta A \times S \cong \text{sk}_0 A \otimes_{\square} S,$$

the transition part can be written using either tensor once the arity object is chosen correctly. But the observation cooperation

$$o : S \longrightarrow O \cdot S$$

forces the output to be constant along execution components.

So there is *not* in general an arbitrary ordinary Moore machine on the vertex set S_0 . A function $S_0 \rightarrow O$ may vary between the two endpoints of an edge, whereas a comodel observation cannot. If one wants arbitrary vertex observations, $S \rightarrow \text{cosk}_0 O$ provides them, but that construction is no longer the ordinary Moore comodel.

This is a useful correction rather than a loss. The Set-based Moore theory asks for a persistent observable attached to the state comodel. The cubical lift reveals that persistence along a genuine execution cell is part of that semantics. A Moore theory in which observations can evolve *during* a higher-dimensional execution should be regarded as a new cubical theory.

14 Levelwise monads: what lifts automatically

The preceding warnings do not mean that ordinary monads cannot be transported to cubical presheaves. Let $T : \mathbf{Set} \rightarrow \mathbf{Set}$ be a finitary or accessible monad and apply it levelwise:

$$(\widehat{T}X)_n = T(X_n).$$

Under the usual hypotheses this levelwise extension carries a canonical strength for the cubical tensor. On a summand corresponding to dimensions p, q , an element $a \in A_p$ and $t \in T(X_q)$ are sent by applying T to the insertion map determined by a : schematically,

$$a \otimes_{\square} t \longmapsto T(\iota_{p,q}(a, -))(t).$$

Naturality with respect to faces in either block is inherited from functoriality of T . Thus “strength with respect to $\otimes_{\square}$ ” is not itself an open problem for these levelwise lifts.

What remains nontrivial is the stronger interaction required by the full Sweedler/duoidal construction: for example, how commutativity of a **Set**-monad transfers to the symmetric cubical tensor, how measuring objects behave, and whether the resulting runners can do more than act blockwise on the value and state dimensions.

This suggests a concrete question which we leave open:

For a levelwise monad T , are natural maps

$$TX \otimes_{\square} S \longrightarrow X \otimes_{\square} S$$

forced to preserve separately the dimensions contributed by X and by S , or can a runner genuinely trade dimension between the value and the hidden state? Even the case $S = \square[1]$ and a small finite monad would be a useful finite test.

15 Is an HDA itself a comodel? Two answers

There are two meanings of the slogan “HDA as comodel,” one nearly tautological and one genuinely interesting.

15.1 The unary presheaf presentation

For any small category C , a presheaf on C can be presented as a comodel in **Set** of the many-sorted unary theory freely generated by C . In particular,

$$\mathbf{PreCub} \simeq \mathbf{Comod}(L_{\square}),$$

for the corresponding many-sorted unary theory. Likewise a labelled HDA can be described through the category of labelled cubes, or the category of elements associated to the label object. This supplies dimension-indexed sorts essentially for free.

But this observation carries little specifically coalgebraic content. For a unary theory, the difference between model and comodel is largely variance, and the presentation does not by itself explain why a 2-cell is a concurrent effect rather than simply one more sort. Nor does it automatically incorporate the chosen initial and accepting subsets I, F .

15.2 The stronger enriched question

The more interesting aim is a theory whose own non-unary operations interact with cubical arities. Power’s enriched Lawvere theories provide a natural template [11]. The symmetric cubical presheaf category is locally presentable, its cubical unit is finitely presentable, and finite cubical objects are stable under the finite dimension-adding tensor constructions suggested by the shuffle formula. This makes it plausible to use finitely presentable cubical objects themselves as arities.

For example, a cooperation of arity $\square[1]$ would have the form

$$S \longrightarrow \square[1] \otimes_{\square} S.$$

Unlike a **Set**-derived cooperation, this map explicitly produces one unit of execution dimension. An arity $\square[2]$ could specify a primitive square, and enrichment could allow coherent higher cells between operations. This is much closer to what we need if the theory itself is to generate HDA geometry.

We should not claim the resulting classification theorem yet. Several points remain to be checked carefully: the exact notion of enriched theory appropriate to the duoidal Sweedler setting, the treatment of initial/final cells and labels, and the relation between symmetric cubes and the newer conclist-based HDA category. But this is the direction in which the structural diagnosis points.

A theorem saying only “precubical sets are comodels of a unary many-sorted theory” repackages the presheaf definition. The real target is a cubically enriched or $\otimes_{\square}$ -linear theory whose operations can create and constrain execution dimensions.

16 Probability and nondeterminism on cubical executions

Part II used nondeterministic and probabilistic residual monads. Their levelwise lifts to cubical presheaves are available, and by the preceding section they carry a natural cubical strength. This gives a conservative first semantics for probabilistic or nondeterministic HDAs: each set of n -cells is transformed by the Set-monad, with face maps transported functorially.

But one must interpret the result correctly. Suppose R is a levelwise probability monad and $\mu \in R(S_2)$ is a distribution over squares. Applying the face maps gives distributions over boundary edges—their marginals. The two composite boundary paths

$$a; b, \quad b; a$$

are not automatically elements of $R(S)$, because path concatenation is extra structure not supplied by the levelwise monad. Thus a random square is coherent as a random *cell*, while its lower-dimensional boundary information survives through the corresponding marginals.

This is already useful, but it is weaker than a probability theory of concurrent executions in which correlations between complete paths, stopping behaviour, or causal histories are primitive. Such a theory may need probability internal to a path/track or PROP-like semantics rather than merely levelwise probability on cells.

The same warning applies to nondeterminism: a set of possible squares and its sets of faces are well defined levelwise, but the relation between whole concurrent histories requires additional execution structure.

17 The candidate Sweedler ambient category

We can now return to the enriched endofunctor category without conflating the two monoidal levels. Let

$$\mathcal{E}_{\square} = \mathbf{sPreCub}$$

with symmetric cubical tensor $\otimes_{\square}$, and consider

$$\boxed{\mathcal{D}_{\square} = \mathcal{E}_{\square}\text{-}\mathbf{End}_{\text{acc}}(\mathcal{E}_{\square})}.$$

A strong/enriched endofunctor has a strength

$$A \otimes_{\square} F(X) \longrightarrow F(A \otimes_{\square} X),$$

so a concurrent execution context may be threaded through the effect. Composition of endofunctors gives one monoidal structure; enriched convolution supplies the natural candidate for the other, as in Parts IV and VI.

The present chapter does not re-prove the full existence theorem. The local-presentability and accessibility arguments from Part VI remain relevant because **sPreCub** is a presheaf category and Day convolution is cocontinuous. What has changed is our interpretation of the monoid objects inside this ambient category. There are now at least two important sectors:

- (a) levelwise/cartesian monads imported from **Set**, whose effects are natural over cubical geometry;
- (b) genuinely cubical-linear monads whose operations and arities use $\otimes_{\square}$ and can manipulate execution dimension.

The second sector is the one in which a runner

$$\theta_X : TX \otimes_{\square} S \longrightarrow R(X \otimes_{\square} S)$$

should carry the strong interpretation originally hoped for in this chapter: the implementation itself respects and transforms genuine higher-dimensional execution geometry.

For sector (a), the same displayed map may exist by strength, but it should not automatically be identified with the ordinary Set-theoretic runner/comodel correspondence. Any such comparison has to use the duoidal interchange and the chosen arity embeddings explicitly.

18 Petri nets: why the linear side now looks inevitable

There is an immediate reason to develop this distinction before turning to Petri nets. Petri nets are already a true-concurrency model. Van Glabbeek gave several translations of nets into HDAs corresponding to different firing interpretations [1, 2]. More recently, Amrane, Bazille, Fahrenberg, Hélouët and Schlehuber-Caissier revisited these translations, including inhibitor arcs and generalized self-modifying nets; the work appeared in the PETRI NETS 2025 proceedings [12].

Part VIII therefore has two complementary routes:

$$\text{Petri nets} \longrightarrow \text{HDA} / \text{cubical execution geometry}$$

and

$$\text{Petri nets} \longrightarrow \text{PROP} / \text{resource-sensitive process geometry}.$$

The present chapter suggests that these should not be regarded as competing encodings. Cubical state is excellent at recording which executions overlap and how their boundaries fit. PROP-like algebra is excellent at representing linear resource flow and parallel composition. The obstruction above tells us that genuinely higher-dimensional effects are likely to live on this linear side rather than inside an unmodified cartesian Lawvere theory.

This gives a useful bridge to the open PROP question in the overall programme: the need for PROP-like structure is not merely imported from Petri-net syntax. It is already visible internally in the failure of cartesian copying and deleting to coexist nontrivially with the cubical dimension tensor.

19 Behavioural equivalence and recent HDA formalisms

The cubical viewpoint also comes with its own behavioural semantics. Open-map methods give history-preserving bisimilarity directly in terms of higher-dimensional paths and unfoldings [8]. More recent work develops path- and cell-bisimulations and modal logics for HDAs in the conlist-based presheaf presentation [9, 10]. These results are relevant to the longer-term **Run** programme because they provide an established target against which a proposed cubical behaviour functor can be tested.

The lesson for us is methodological. We should not define “behaviour of a cubical runner” merely by forgetting to its 1-skeleton and then applying ordinary automata theory. A satisfactory behaviour construction should retain enough higher-dimensional information to compare with history-preserving, cell-based, or related true-concurrency equivalences. The precise relation to the Beh-to-profunctor programme of Part IX remains open, but there is now a concrete external benchmark.

20 Toward a Sweedler calculus of concurrency

Part VI suggested that changing the ambient category can turn resource accounting into a calculus: local potential inequalities compose to give global amortised bounds. The corresponding ambition in the cubical setting is not primarily to count resources but to reason compositionally about *progress*. Can local information about the possible executions of components be combined to prove that a wired system cannot deadlock, or that specified actions remain independent?

Higher-dimensional automata already provide geometric tools for such questions. Deadlocks, forbidden regions and spaces of executions can be read from a directed cubical state space [13]. What the runner formalism might add is a compositional layer: a component should expose enough of its concurrency behaviour at its boundary that properties of a composite can be inferred without reconstructing its entire global state space.

The smallest useful test is the classical two-process, two-lock deadlock.

20.1 Two processes and two locks

Let a and b be mutexes, each of capacity one. Write P_r for acquisition of r and V_r for its release. Consider the two sequential processes

$$P = P_a; P_b; V_b; V_a, \quad Q = P_b; P_a; V_a; V_b.$$

Thus P acquires a before b , while Q acquires them in the opposite order.

Before imposing the mutex constraints, the independent composition of the two process paths has cubical state space obtained from their cubical tensor. In particular, at the initial state s , the actions P_a and P_b performed by the two different processes are independent. They therefore determine a square

$$\begin{array}{ccc} s & \xrightarrow{P_a} & p \\ Q_b \downarrow & & \downarrow Q_b \\ q & \xrightarrow{P_a} & d. \end{array}$$

Here p is the state in which P holds a , q the state in which Q holds b , and d the state in which P holds a while Q holds b . The square is important: the two first acquisitions really are independent.

Cubical semantics remembers this instead of representing their two interleavings as unrelated paths.

The problem appears at the far vertex d . Process P next requests b , but b is held by Q ; process Q next requests a , but a is held by P . Hence neither process can move:

$$d \not\stackrel{P_b}{\rightarrow} -, \quad d \not\stackrel{Q_a}{\rightarrow} -.$$

If d is not an accepting state, it is a deadlock.

There is a useful local geometric formulation. For a vertex x of an HDA X , let

$$\text{Lk}_X^+(x)$$

denote its future link: vertices of the link are elementary actions enabled at x , edges record pairs which can proceed independently, and higher simplices record higher compatible families. At the deadlock,

$$\text{Lk}_X^+(d) = \emptyset.$$

Thus deadlock is the zero-dimensional end of a more general hierarchy. A future link may be nonempty but lose edges or higher simplices, recording a loss of available concurrency before progress disappears altogether.

This already suggests a local progress judgement

$$X, x \models \text{progress} \iff \text{Lk}_X^+(x) \neq \emptyset$$

for every reachable non-final state x . But checking this directly on the global HDA still requires constructing the global state space. A Sweedler calculus should do better.

20.2 A compositional summary: the lock-order relation

Each individual process contains a very small amount of information which already predicts the problem. For a process M , define its lock-order relation

$$\Lambda(M) \subseteq L \times L$$

by

$$r \prec_M s$$

when some local execution of M can request s while still holding r . For the two processes above,

$$\Lambda(P) = \{a \prec b\}, \quad \Lambda(Q) = \{b \prec a\}.$$

Composition exposes the obstruction:

$$\Lambda(P) \cup \Lambda(Q) = \{a \prec b, b \prec a\}.$$

The combined relation contains a directed cycle.

This is the familiar lock-order argument, but it has exactly the shape we would like from a compositional semantics. Instead of exploring the product state space, each open component exports a small contract, and composition combines contracts by a simple algebraic operation.

Under the elementary assumptions of the present example this gives a genuine soundness criterion.

Lock-order criterion. Consider finitely many sequential processes interacting only through blocking mutex acquisition and release, and suppose that a nonterminated process can be blocked only while attempting to acquire a mutex. If the union

$$\Lambda = \bigcup_i \Lambda(P_i)$$

of their lock-order relations is acyclic, then no reachable global state is a deadlock caused by circular lock waiting.

Indeed, suppose a deadlocked state existed. Every nonterminated process would be waiting for a lock held by another process. Following ownership and waiting around the finite collection of processes eventually produces a cycle

$$r_0 \prec r_1 \prec \cdots \prec r_n = r_0.$$

Every edge occurs in the lock-order relation of one of the processes, so it gives a cycle in Λ , contradicting acyclicity.

The criterion is sufficient rather than complete: cyclic acquisition orders need not force a deadlock on every execution. This asymmetry is desirable. We are looking for a compositional certificate which can be checked before the full state space is generated, not merely a different presentation of exhaustive reachability.

20.3 Repairing the example

The standard repair is to impose a common lock order. Replace Q by

$$Q' = P_a; P_b; V_b; V_a.$$

Then

$$\Lambda(P) = \Lambda(Q') = \{a \prec b\},$$

and hence

$$\Lambda(P) \cup \Lambda(Q') = \{a \prec b\}$$

is acyclic.

Notice what has changed geometrically. At the initial state, the two attempts to acquire a are now competing rather than independent: because a is a mutex, they do not span a filled concurrency square. One process wins the lock. Once it holds a , it can acquire b , finish its critical section and release the locks, after which the other can proceed. Competition remains, but circular waiting has disappeared.

This distinction is precisely what a higher-dimensional semantics ought to preserve:

competition is not deadlock, and interleaving is not independence.

The 1-skeleton tells us which transitions are possible. The higher cells tell us which possible transitions commute as genuinely independent activity. The future link tells us how much local concurrency remains available.

20.4 Where the Sweedler structure should enter

So far the example is standard concurrency theory. The proposed Sweedler contribution is to make the passage

$$\text{component} \mapsto \text{progress contract}$$

compatible with the ways in which components are wired.

The lock-order relation gives the smallest prototype. Let

$$\text{Rel}(L)$$

denote the commutative idempotent monoid of relations on the lock set L , with multiplication given by union. We would like an abstraction

$$\Lambda : \{\text{open cubical processes over } L\} \longrightarrow \text{Rel}(L)$$

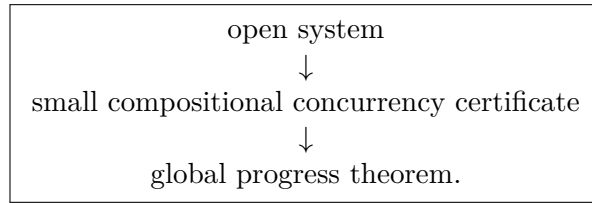
such that independent or parallel composition satisfies, in the simple lock discipline,

$$\Lambda(P \otimes Q) = \Lambda(P) \cup \Lambda(Q).$$

For more general forms of wiring, equality may be too strong: synchronization can create dependencies not visible in either component separately. The appropriate target may therefore be a lax law

$$\Lambda(P) \cup \Lambda(Q) \subseteq \Lambda(P \parallel Q).$$

The point is not the particular relation Λ . It is the pattern:



This is where runners may become useful. An open process does not live in isolation: its abstract commands are realised against a residual environment containing locks, shared state, schedulers or other processes. Schematically one expects a runner

$$T_{\text{proc}} \rightsquigarrow R_{\text{env}}$$

whose state object is itself cubical. Wiring runners then composes open implementations. A concurrency analysis should be an invariant, predicate or algebra on such runners which is respected by this composition.

The revised distinction between cartesian and genuinely cubical theories is essential here. A Set-derived lock theory lifted pointwise to cubical sets can label the cells of an already existing HDA, but by itself it cannot account for the creation and destruction of execution dimensions. A genuine theory of concurrency should have operations or cooperations native to $\otimes_{\square}$. At the most elementary level, a cooperation

$$S \longrightarrow \square[1] \otimes_{\square} S$$

creates a direction of execution. Higher cubical arities can express compatible families of directions. Acquisition, synchronization and joining should then be able to alter the future-link geometry rather than merely act independently at every dimension.

The two-lock example suggests what such a theory has to explain. Before resource constraints are imposed, the first actions of P and Q produce a square. Wiring both processes through the shared mutex environment restricts the available continuation at its far vertex until

$$\text{Lk}^+(d) = \emptyset.$$

The deadlock is therefore not merely a bad vertex. It is produced by interaction between the cubical behaviour of the processes and the residual resource environment. A successful Sweedler formulation should make that interaction functorial.

20.5 From deadlock freedom to a progress calculus

The lock-order relation is deliberately coarse. It forgets almost all of the HDA, but this makes it a useful first compositional abstraction. The next stages should retain progressively more geometry.

At a vertex x , one can record not just whether the future link is empty but its dimension:

$$c(x) = \begin{cases} 0, & \text{Lk}^+(x) = \emptyset, \\ 1 + \dim \text{Lk}^+(x), & \text{otherwise.} \end{cases}$$

This gives a crude measure of the maximum amount of locally available independent activity. More invariantly, one may retain the homotopy or homology type of the future link. The successive questions are then

certificate	question
$\text{Lk}^+(x) \neq \emptyset$	can the system make progress?
$\dim \text{Lk}^+(x)$	how much concurrency remains locally?
$\pi_0 \text{Lk}^+(x)$	are the available futures separated into alternatives?
$H_* \text{Lk}^+(x)$	what higher obstruction structure remains?

Globally, directed path spaces

$$\vec{P}_X(x, y)$$

refine reachability in the same fashion: nonemptiness asks whether y is reachable from x , while connected components distinguish essentially different families of schedules.

This suggests a hierarchy of possible calculi. The first is the finite lock-order analysis just described. A second would attach future-link information to interfaces. A third would attach homotopy-invariant information about spaces of executions. In each case the central problem is the same:

Find an abstraction of cubical behaviour which is strong enough to prove progress properties but sufficiently algebraic to be transported along runners and composed under wiring.

This is the concurrency analogue of the amortisation problem of Part VI. There the aim is to replace a global resource calculation by local potentials which compose. Here the aim is to replace global exploration of an HDA by local progress certificates which compose.

20.6 A first research target

The two-lock example therefore gives a concrete first target for the theory. We would like to construct:

1. a cubical theory T_{lock} of acquisition, release and synchronization;
2. a residual runner semantics which sends open lock-using processes to HDAs;
3. a compositional abstraction

$$\Lambda : \mathbf{Run}(T_{\text{lock}}, R) \longrightarrow \mathbf{Rel}(L)$$

or its appropriate categorical refinement;

4. a proof that composition of runners maps to union, or to a controlled lax extension, of lock dependencies;
5. a soundness theorem saying that acyclicity of the resulting dependency relation implies absence of reachable lock deadlocks.

Even this modest theorem would be significant. It would show that the Sweedler structure is doing more than providing another semantics for concurrency: it would turn the interface and composition structure of runners into a verification method.

There is then an obvious bridge to the next part. Petri nets combine the two themes which Parts VI and VII isolate. They have genuine resource accounting—tokens are consumed and produced—but independent transitions also generate genuine concurrent geometry. The eventual hope is therefore not two unrelated calculi, one for amortisation and one for deadlock, but a single compositional framework in which quantitative resource potential and higher-dimensional progress can interact.

21 What Part VII has established, and what remains open

Established or elementary background

- Precubical and symmetric cubical objects provide presheaf geometries for higher-dimensional execution.
- The dimension-adding Day tensor satisfies $\square[p] \otimes_{\square} \square[q] \cong \square[p+q]$; after symmetrising, the cell formula acquires shuffle factors.
- Ordinary ordered precubical sets are not braided under this tensor, so a symmetric cubical base is the natural choice for the present symmetric Sweedler architecture.
- The pair $(\otimes_{\square}, \times)$ is canonically duoidal but non-normal, with units $\square[0]$ and $\Delta 1$.
- For an ordinary Set-based Lawvere theory T , cubical comodels are presheaves of Set-comodels:

$$\mathbf{Comod}_T(\mathbf{PreCub}) \simeq [\square^{\text{op}}, \mathbf{Comod}_T(\mathbf{Set})].$$

- Comonoids for the cubical tensor are concentrated in dimension zero. Hence ordinary reader-like copying/deleting structure does not extend nontrivially to higher cubical dimensions.

- Levelwise Set-monads carry a canonical strength for $\otimes_{\square}$.
- Every presheaf category can be presented by a many-sorted unary theory, so “HDA as comodel” is formally easy at that unary level but does not yet explain concurrency as an effect.

The research direction now isolated

- Develop $\otimes_{\square}$ -linear or cubically enriched theories whose operations can create and consume execution dimensions.
- Determine the exact Sweedler measuring hypotheses for the enriched accessible endofunctor category on the chosen cubical base.
- Compare symmetric precubical sets with the newer conclist-based category of HDAs.
- Incorporate labels and initial/final cells into a fibred or parameterized theory rather than adding them externally.
- Test whether natural $\otimes_{\square}$ -runners for levelwise monads can transfer dimension between value and state, or whether they are forced to be blockwise.
- Develop probability and nondeterminism at the level of complete concurrent executions, not only levelwise cells and their marginals.
- Compare cubical behavioural semantics with open-map, history-preserving, path-, and cell-bisimulation notions.
- Relate the cubical-linear theory directly to the PROP/resource semantics of Petri nets in Part VIII.

22 The larger lesson

Parts V–VII have changed the ambient world in increasingly structural ways:

base	what state remembers
Set	points and ordinary hidden state
V-Pred	certificates, bounds, or truth values
C-Set	resource grades and stored potential
PreCub / sPreCub	higher-dimensional execution geometry.

But Part VII adds a second lesson. Changing the base is not by itself enough. A Set-derived effect may live naturally on cubical objects while remaining algebraically blind to the fact that dimensions can be added. To make concurrency part of the effect rather than merely part of its state space, the arities of the theory must change as well.

Part VII in one sentence. Cubical state lets a runner carry hidden execution geometry, but ordinary cartesian effects only travel along that geometry; effects which genuinely manipulate concurrency should be sought among theories enriched or linear in the dimension-adding cubical tensor.

That conclusion is more restrictive than the first version of the chapter, but also more useful. It tells us exactly where the next construction must occur, and why Petri nets and PROPs are not a detour from the Sweedler programme but one of its natural higher-dimensional continuations.

References

References

- [1] R. J. van Glabbeek. On the expressiveness of higher dimensional automata. *Theoretical Computer Science* 356(3):265–290, 2006. doi:10.1016/j.tcs.2006.02.012.
- [2] R. J. van Glabbeek. Erratum to “On the expressiveness of higher dimensional automata”. *Theoretical Computer Science* 368(1–2):168–194, 2006. doi:10.1016/j.tcs.2006.06.024.
- [3] U. Fahrenberg. A category of higher-dimensional automata. In *Foundations of Software Science and Computation Structures (FoSSaCS 2005)*, LNCS 3441, pp. 187–201. Springer, 2005.
- [4] M. Grandis and L. Mauri. Cubical sets and their site. *Theory and Applications of Categories* 11:185–211, 2003.
- [5] M. Grandis. The role of symmetries in cubical sets and cubical categories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques* 50(2):102–143, 2009.
- [6] U. Fahrenberg, C. Johansen, G. Struth, and K. Ziemiański. Languages of higher-dimensional automata. *Mathematical Structures in Computer Science* 31(5):575–613, 2021. arXiv:2103.07557.
- [7] T. Kahl. Topological abstraction of higher-dimensional automata. *Theoretical Computer Science* 631:97–117, 2016. doi:10.1016/j.tcs.2016.04.018; arXiv:1506.02270.
- [8] U. Fahrenberg and A. Legay. History-preserving bisimilarity for higher-dimensional automata via open maps. *Electronic Notes in Theoretical Computer Science* 298:165–178, 2013. doi:10.1016/j.entcs.2013.09.012; arXiv:1209.4927.
- [9] S. Zouari, K. Ziemiański, and U. Fahrenberg. Bisimulations and logics for higher-dimensional automata. In *Theoretical Aspects of Computing (ICTAC 2024)*, LNCS, pp. 132–150. Springer, 2025. arXiv:2402.01589.
- [10] S. Zouari, R. van Glabbeek, and K. Ziemiański. Bisimulations and modal logics for higher dimensional automata. In *37th International Conference on Concurrency Theory (CONCUR 2026)*, LIPIcs 391, 49:1–49:19, 2026. doi:10.4230/LIPIcs.CONCUR.2026.49.
- [11] J. Power. Enriched Lawvere theories. *Theory and Applications of Categories* 6(7):83–93, 1999.
- [12] A. Amrane, H. Bazille, U. Fahrenberg, L. Hélouët, and P. Schlehuber-Caissier. Petri Nets and Higher-Dimensional Automata. In *Application and Theory of Petri Nets and Concurrency: PETRI NETS 2025*, pp. 18–40. Springer, 2025. arXiv:2502.02354.
- [13]