

Getting to Grips with Sweedler Runners for Systems

A conversational tutorial, Part XI: closing runs
Automata, trace languages, wheeled props, Doi–Peliti semantics, and closing runs

September 2026

Aim. The runners of earlier parts have mostly described *open* execution: one theory is implemented in another while hidden state is carried along. Part XI asks what changes when some boundary is closed. We first use the theorem of Spivak, Schultz and Rupel that traced string diagrams are labelled oriented one-dimensional cobordisms to make the geometry of closure precise. Gustafson et al. then show how an automaton evaluates decorated intervals and circles, giving open and circular behaviours. This leads naturally to the shadow of **Prof**: if $\text{Beh} : \mathbf{Run}(\mathcal{D}) \rightarrow \mathbf{Prof}$ sends a theory to its behaviour category, then the shadow of an endoprofunctor is a canonical receptacle for cyclic behaviour. We then pass from the $(1,1)$ -valent automaton case to whole-grain Petri nets and free wheeled props, before asking how Doi–Peliti operator semantics and Sweedler runners interact with graphical contraction. The central caution remains that $\mathbf{Run}(\mathcal{D})$ does *not* automatically come with a trace.

Contents

1	Why traces enter the story	2
1.1	Why traced string diagrams are cobordisms	3
2	Automata as the first complete example	3
2.1	Open behaviour: the decorated interval	4
2.2	Closing the interval: circular behaviour	4
2.3	From loop behaviour to the shadow of Prof	5
2.4	The path-integral viewpoint	8
2.5	Automata as the 1-in, 1-out fragment	8
3	From a net to its wheeled process theory	9
3.1	Raynor’s free wheeled prop	9
3.2	What does a wheel mean?	10
3.3	A syntactic presentation: contraction of indices	10
4	Doi–Peliti semantics and internal circulation	11
4.1	A catalytic strand	11
4.2	Do not identify the number operator with the trace	12
4.3	Beyond static catalysts	12
4.4	Stochastic dynamics	12

5	Three meanings of closing	13
5.1	Representing the free wheeled prop	14
5.2	From cobordisms to wheeled graphs	14
6	Where do runners enter?	14
6.1	A compatibility question for Run	15
6.2	Should Beh preserve the shadow?	15
6.3	A regulator as an internal cycle	16
7	Enriched shadows and resource-valued closed behaviour	16
7.1	The cost-enriched case	17
7.2	Shadow first, valuation second	18
7.3	From Boolean traces to cyclic invariants	19
8	Refinement and layered wheeled props	19
8.1	Zooming in and out	19
8.2	Refinement and layered wheeled props	20
8.3	Weighted refinement and path sums	20
9	Loops, field theory, and the remaining question	21
10	Relation to higher-dimensional automata	22
11	What Part XI has added	22

1 Why traces enter the story

Up to this point we have mostly thought of a run as something with a beginning and an end. A theory supplies operations, a comodel or runner realises them on a state space, and wiring allows several such pieces to be assembled into a larger system. There is, however, another operation which appears repeatedly in systems theory:

take part of the output of a process
and feed it back into part of its input.

In a traced symmetric monoidal category this has the familiar form

$$f : X \otimes U \longrightarrow Y \otimes U \quad \rightsquigarrow \quad \mathrm{Tr}^U(f) : X \longrightarrow Y.$$

The object U has disappeared from the external interface. It has become internal to the resulting process.

There are several reasons why this matters for us. Feedback is already ubiquitous in ordinary systems: regulators, controllers, circuits and recursively organised processes all close interfaces internally. Petri nets and graphical process theories have an especially natural notion of closing a wire. Once an output and input of matching type are identified, one obtains a wheel. Finally, once a process theory is represented linearly, closed pieces may acquire numerical or operator-theoretic values: traces, partition functions, loop weights, transfer operators and, in the stochastic setting, Doi–Peliti-style expressions.

Before turning to automata or Petri nets, however, there is a structural result which tells us why one-dimensional cobordisms should appear at all.

1.1 Why traced string diagrams are cobordisms

The appearance of oriented one-dimensional cobordisms is not merely a suggestive way of drawing feedback. Spivak, Schultz and Rupel [5] show that labelled oriented 1-cobordisms give a complete geometric presentation of traced symmetric monoidal structure. If $\mathcal{O}$ is a set of wire labels, they form an operad

$$\mathbf{Cob}_{/\mathcal{O}}$$

of oriented one-dimensional cobordisms whose strands are labelled in $\mathcal{O}$. Their main result says, in particular, that the axioms of traced symmetric monoidal categories are fully encoded by this cobordism operad: $\mathbf{Cob}_{/\mathcal{O}}$ -algebras, as the label set varies, recover traced categories with varying object sets. Thus the familiar traced diagram

$$f : X \otimes U \longrightarrow Y \otimes U, \quad \mathrm{Tr}^U(f) : X \longrightarrow Y,$$

should be read literally as a one-dimensional cobordism. The external wires give the boundary, while the U -output is glued back to the matching U -input and becomes internal to the cobordism.

The simplest closed connected component is therefore a circle. This is already enough to explain why a trace has no preferred point at which the feedback loop begins: once the strand has been closed, moving the cut around the circle does not change the cobordism. The same cyclicity will reappear algebraically when we reach the shadow of **Prof**.

This theorem gives the geometric syntax. The paper of Gustafson, Im, Kaldawy, Khovanov and Lihn [6] supplies an unusually concrete systems-theoretic semantics for it. Finite automata evaluate defect-decorated intervals and circles, so that the passage from open to closed cobordism becomes a passage from ordinary accepted behaviour to loop behaviour.

2 Automata as the first complete example

Let Q be a nondeterministic finite automaton over an alphabet Σ . Gustafson et al. associate to Q a one-dimensional oriented TQFT with inner endpoints and zero-dimensional defects labelled by letters of Σ . The target is the category of finite free modules over the Boolean semiring

$$\mathbb{B} = \{0, 1\}, \quad 1 + 1 = 1.$$

The positively oriented point is assigned the free Boolean module

$$F_Q(+) = \mathbb{B}^Q,$$

whose basis is the set of states of the automaton. A letter $a \in \Sigma$, viewed as a defect on an oriented strand, acts by the Boolean transition operator

$$m_a : \mathbb{B}^Q \longrightarrow \mathbb{B}^Q.$$

For a word $\omega = a_1 \cdots a_n$, the corresponding decorated strand carries the composite operator m_ω , with the order determined by the orientation convention.

This is already strikingly close to the way we have been talking throughout the tutorial. The automaton has a state space. Letters act by state transitions. A decorated one-manifold records a possible piece of execution. What the TQFT adds is a disciplined way of asking what happens when the endpoints of that execution are left open or are glued together.

2.1 Open behaviour: the decorated interval

The first test object is an interval decorated by the word ω . The two inner endpoints encode the initial and accepting conditions of the automaton. Its Boolean value is

$$1 \iff \omega \in L(Q),$$

where $L(Q)$ is the ordinary accepted language. In other words, the interval asks whether there exists a path through the automaton, labelled by ω , beginning in an initial state and ending in an accepting state.

This is a very concrete version of a behaviour map. The automaton presents a stateful system; the decorated interval is an experiment with exposed boundary conditions; the result records whether the corresponding open run exists. It is useful to write this suggestively as

$$\text{Beh}_I(Q)(\omega) = \begin{cases} 1, & \text{if an accepting } \omega\text{-run exists,} \\ 0, & \text{otherwise.} \end{cases}$$

Nothing profound depends on the notation Beh_I . The point is simply that the ordinary language of the automaton is already a boundary-sensitive behaviour semantics.

2.2 Closing the interval: circular behaviour

Now glue the two ends of the decorated interval together. Geometrically we obtain a circle carrying the same cyclically ordered defects. Algebraically the TQFT assigns the trace of the corresponding Boolean endomorphism:

$$\text{tr}_{\mathbb{B}Q}(m_\omega).$$

Over the Boolean semiring this value is 1 exactly when there is some state $q \in Q$ from which one can follow a path labelled by ω and return to q . Thus the circle detects a closed run:

$$\text{Beh}_o(Q)(\omega) = \begin{cases} 1, & \text{if there is a closed } \omega\text{-path,} \\ 0, & \text{otherwise.} \end{cases}$$

Gustafson et al. call the corresponding collection the trace or circular language of the automaton. The important point for us is that this is not determined merely by the accepted interval language. Two automata can accept the same regular language while giving different values on decorated circles.

So even in this elementary case there are two natural kinds of observable behaviour:

open behaviour	closed behaviour
$q_{\text{in}} \xrightarrow{\omega} q_{\text{out}}$	$q \xrightarrow{\omega} q$
interval	circle.

The change is not merely that we have drawn the same computation in a different shape. Closing the boundary changes the class of experiments we allow. Open behaviour singles out designated initial and final conditions; circular behaviour instead asks whether the state evolution can be made globally consistent after the two ends are identified.

This is a useful refinement of the Beh story from earlier parts. Behaviour is not a single undifferentiated object. It depends on the contexts in which the machine is allowed to be tested. A trace enlarges the class of tests by allowing outputs to be reattached to matching inputs.

2.3 From loop behaviour to the shadow of Prof

There is a categorical construction which packages precisely this passage from open behaviour to cyclic behaviour. In the **Set**-based accessible case the behavioural assignment can be made double-functorial. For each theory T , Garner associates a behaviour category $\mathcal{B}_T$, with

$$T\text{-}\mathbf{Comod} \simeq [\mathcal{B}_T, \mathbf{Set}].$$

A tight interpretation of theories induces a cofunctor between the corresponding behaviour categories, while a runner induces a profunctor. Thus, after fixing the variance convention once and for all, there is a behavioural double functor

$$\mathbf{Beh} : \mathbf{Run}(\mathcal{D}) \longrightarrow \mathbf{CofProf}, \quad \mathcal{D} = [\mathbf{Set}, \mathbf{Set}]_{\text{acc}},$$

where **CofProf** denotes the double category whose objects are small categories, whose tight arrows are cofunctors, and whose loose arrows are profunctors. Its horizontal bicategory is the usual bicategory **Prof**.

The loose part of this construction comes directly from the action of runners on comodels. If

$$M : T \rightsquigarrow R$$

has residual state S , then horizontal composition with M sends an R -comodel to a T -comodel:

$$M^* : R\text{-}\mathbf{Comod} \longrightarrow T\text{-}\mathbf{Comod}.$$

On underlying carriers this is $S \times -$. Since colimits of comodels are created on the underlying sets and $S \times -$ preserves colimits, M^* is cocontinuous. Transporting along Garner's equivalences gives a cocontinuous functor

$$[\mathcal{B}_R, \mathbf{Set}] \longrightarrow [\mathcal{B}_T, \mathbf{Set}],$$

and hence, by density, a representing profunctor

$$P_M : \mathcal{B}_R \nrightarrow \mathcal{B}_T.$$

Explicitly,

$$P_M(t, r) = M^*(\mathcal{B}_R(r, -))(t).$$

For composable runners

$$T \xrightarrow{M} R \xrightarrow{N} U,$$

horizontal composition becomes profunctor composition:

$$P_{N \odot M}(t, u) \cong \int^{r \in \mathcal{B}_R} P_M(t, r) \times P_N(r, u).$$

Thus the coend which defines profunctor composition is not being added by hand: it is the behavioural image of horizontal composition in **Run**($\mathcal{D}$).

The tight part comes from Garner's functoriality of behaviour categories. A theory map

$$f : T \longrightarrow R$$

induces a cofunctor between the corresponding behaviour categories. The direction is contravariant at the level of comodel semantics, since restriction along f sends an R -comodel to a T -comodel.

After choosing the orientation convention in **CofProf**, this fits together with the profunctorial loose part above. Runner cells are sent to the corresponding cells between profunctors, so that the behavioural assignment respects the full double-category structure rather than only horizontal composition.

This is exactly the structure needed to import the canonical shadow of **Prof** [4]. For an endoprofunctor

$$P : \mathcal{A} \rightharpoonup \mathcal{A}$$

its shadow is

$$\langle P \rangle = \int^{a \in \mathcal{A}} P(a, a).$$

Consequently an endo-runner

$$M : T \rightsquigarrow T$$

has a canonical behavioural shadow

$$\boxed{\langle M \rangle_{\text{Beh}} = \int^{b \in \mathcal{B}_T} P_M(b, b).}$$

The cyclicity isomorphism in **Prof**,

$$\langle P \odot Q \rangle \cong \langle Q \odot P \rangle,$$

therefore gives the corresponding cyclicity for behavioural images of composable runners. In this sense the shadow structure needed here does not have to be constructed independently inside **Run**($\mathcal{D}$): it is transported from **Prof** along the horizontal part of Beh.

The special case of the identity runner recovers the familiar expression

$$\langle 1_{\mathcal{B}_T} \rangle = \int^{x \in \mathcal{B}_T} \mathcal{B}_T(x, x).$$

Before taking the coend, an element of $\mathcal{B}_T(x, x)$ is a loop based at x . The coend removes the privileged cut. Thus, if

$$x \xrightarrow{f} y \xrightarrow{g} x,$$

then

$$gf : x \rightarrow x, \quad fg : y \rightarrow y$$

represent the same cyclic behaviour in the shadow. Algebraically this is the analogue of sliding the cut around a closed oriented circle.

The automaton example makes this particularly concrete, but one should distinguish the universal behaviour category of a theory from the path category of a particular automaton. For the free unary theory

$$W_\Sigma = F(\Sigma y),$$

Garner's universal behaviour category is the one-object category

$$\mathcal{B}_{W_\Sigma} \simeq \mathbf{B}\Sigma^*,$$

whose endomorphism monoid is Σ^* . A particular nondeterministic automaton

$$(Q, \delta, Q_{\text{in}}, Q_t)$$

then gives a representation of this universal behaviour category: each word

$$w \in \Sigma^*$$

acts by the relation

$$R_w \subseteq Q \times Q$$

of w -labelled paths, or equivalently by the corresponding Boolean-linear map

$$T_w : \mathbb{B}Q \longrightarrow \mathbb{B}Q.$$

For a fixed word w , regard R_w as an endoprofunctor on the discrete category Q :

$$P_w : Q \nrightarrow Q.$$

Its shadow is

$$\langle P_w \rangle = \int^{q \in Q} P_w(q, q).$$

Since Q is discrete, this reduces to

$$\coprod_{q \in Q} P_w(q, q),$$

the set of closed w -runs together with their chosen base states. Applying Boolean existential valuation gives a value in $\mathbb{B}$, equal to 1 precisely when the automaton admits a closed w -labelled run. This is the Boolean trace appearing in the automaton TQFT of Gustafson et al. The **Prof** shadow therefore separates two operations which are combined in the Boolean matrix trace:

closed runs $\longrightarrow$ cyclic behaviour $\longrightarrow$ Boolean value.

For a non-discrete behaviour category the coend does more than take a disjoint union of based loops: it also quotients by the actions of behaviour morphisms. It therefore supplies a canonical notion of unbased or cyclic behaviour. The point is not merely that loops exist in the behaviour category, but that the shadow identifies the different cuts through which one and the same closed behaviour can be presented.

This gives a useful hierarchy:

runner $M : T \rightsquigarrow T$ $\downarrow$ Beh $P_M : \mathcal{B}_T \nrightarrow \mathcal{B}_T$ $\downarrow$ shadow $\int^{b \in \mathcal{B}_T} P_M(b, b)$ cyclic behaviour $\downarrow$ valuation Boolean, numerical, enriched, or TQFT value.
--

This also clarifies the relation with the feedback trace discussed earlier. A partial trace on runners closes a chosen feedback interface,

$$T + X \rightsquigarrow R + X \quad \longmapsto \quad T \rightsquigarrow R,$$

when the corresponding feedback is traceable. The **Prof** shadow performs a different operation: it takes an endo-runner and passes from based to cyclic behaviour. The two constructions need not be identified. For the purposes of the present Part, the latter may already provide the essential trace-like structure: once an endo-runner has been sent to an endoprofunctor, the canonical shadow of **Prof** supplies the cyclic closure needed for comparison with wheeled diagrams and one-dimensional TQFTs.

This puts the cobordism and profunctor pictures remarkably close together. Spivak–Schultz–Rupel close a labelled interval geometrically; the shadow closes profunctorial behaviour by a coend. In both cases a closed loop has no distinguished starting point. The behavioural double functor makes this comparison structural rather than merely analogical:

$$\mathbf{Run}(\mathcal{D}) \longrightarrow \mathbf{CofProf}$$

with horizontal bicategory **Prof**, followed by the canonical shadow of **Prof**. The remaining valuation determines what information about a closed run is retained: mere existence in the Boolean case, but potentially multiplicity, weight, probability, cost, homology class, or other enriched data in more general settings.

2.4 The path-integral viewpoint

Gustafson et al. also emphasise that both the accepted language and the TQFT admit a path-integral interpretation. For an interval one sums, in the Boolean sense, over state histories compatible with the word and the endpoint conditions. For a circle one sums over compatible closed state histories. Boolean addition means that the sum only remembers whether at least one such history exists.

This gives us a very simple prototype for the stochastic constructions later in the chapter:

Boolean automaton: sum over finite state histories $\downarrow$ stochastic net: weighted sum over population histories.

In the first case the weights live in $\mathbb{B}$. In a weighted or linear automaton they may live in a ring or semiring. In Doi–Peliti theory the state basis is no longer a finite set of control states but the set of occupation-number configurations, and the weights come from stochastic rates.

2.5 Automata as the 1-in, 1-out fragment

Why should this example be relevant to Petri nets? Graphically, the answer is simple. Every primitive transition of an automaton has one state on the incoming side and one state on the outgoing side. The generating vertices are therefore of valence $(1, 1)$. Connected process diagrams built from them are oriented intervals and circles, decorated by transition labels. This is precisely the dimension in which the Spivak–Schultz–Rupel cobordism presentation applies most literally, and the Prof shadow gives the corresponding passage from based paths to cyclic behaviour.

A Petri-net transition, by contrast, may have an arbitrary finite input profile and an arbitrary finite output profile:

$$M \longrightarrow N.$$

Once (m, n) -valent vertices are admitted, intervals and circles are no longer enough; branching and merging force us into general directed graphs. Correspondingly, traced-category syntax enlarges to

the wheeled-prop syntax appropriate to arbitrary graphical processes. The progression is therefore not merely from small machines to larger ones, but from the one-dimensional geometry of trace to its many-input/many-output graphical analogue:

automata	$\subset$	Σ -nets / Petri nets
$(1, 1)$ -vertices		(m, n) -vertices
oriented intervals and circles		directed graphs and wheels
traced categories / Prof shadow		wheeled props.

The automaton case therefore gives us a complete test case: geometrically a circle, behaviourally a closed run, and profunctorially a shadow. We can now ask for the corresponding structure in the general graphical setting.

3 From a net to its wheeled process theory

Recall the whole-grain view of Petri nets. Rather than recording a transition merely by multiplicities of input and output places, one keeps its individual input and output arcs visible. A transition requiring two tokens of the same place therefore has two distinguishable input flags. This is exactly the graphical viewpoint supplied by digraphical species, also known as oriented graphical species and, in the terminology of Baez and collaborators, Σ -nets.

A whole-grain Petri net may be regarded as a flat digraphical species. Flatness says, roughly, that the symmetric-group actions on the input and output flags are free. Multiplicity is therefore recovered by counting fibres rather than being inserted beforehand as a numerical decoration.

Starting with such a net P , we should distinguish the generating data from the process theory freely generated by it. At the generating level we have primitive transitions

$$t : M \longrightarrow N$$

between finite families of places. We may compose compatible transitions sequentially, compose independent processes in parallel, and use identities and symmetries. The important new point is that the natural completion does not have to stop there.

3.1 Raynor's free wheeled prop

Raynor's construction [11, 12] gives a particularly useful way of organising this completion. In the oriented setting there is a composite monad on oriented graphical species which we shall write

$$\mathbb{O}_{\text{or}} = L_{\text{or}} D_{\text{or}} T_{\text{or}}.$$

Its stages have a useful operational reading. The monad T_{or} generates graphical substitution and hence sequential composite processes. The monad D_{or} adjoins the appropriate units, giving wires which simply pass a place through unchanged. The monad L_{or} adjoins disjoint union, giving parallel composition and the symmetric monoidal structure. Their composite produces the free wheeled prop generated by P :

$$\mathbb{O}_{\text{or}} P.$$

Besides sequential and parallel composition, outputs and inputs carrying the same colour may now be contracted. Schematically,

$$f : X \otimes U \longrightarrow Y \otimes U \quad \rightsquigarrow \quad \text{Tr}^U(f) : X \longrightarrow Y.$$

Graphically, the U -output has been joined back to the U -input. Thus a Petri net does not merely generate open firing processes. It generates a syntax in which parts of a process may circulate internally.

We may summarise the passage as

$$P \longrightarrow \text{Proc}(P) \longrightarrow \text{Wheel}(P) \simeq \mathbb{O}_{\text{or}} P.$$

The distributive laws in Raynor’s construction matter because sequential composition, parallel composition, units and contraction are not four unrelated operations. They interact coherently inside a single process calculus.

3.2 What does a wheel mean?

At this stage one should resist interpreting a wheel too quickly. A wheel is first of all a *graphical contraction*: an output wire and an input wire of matching colour have been identified. Different semantics may subsequently read this as feedback, a hidden internal resource, a cyclic process, a contraction of tensor indices, a contribution to a partition function, or eventually an operator trace. These are interpretations of the graphical operation; they should not be put into its definition.

The automaton example makes this caution concrete. There, gluing the two ends of a decorated interval produces a circle and the TQFT evaluates that circle by a trace. For a general wheeled prop the contraction exists already at the level of syntax, before any particular Boolean, linear, stochastic or TQFT-style valuation has been chosen.

3.3 A syntactic presentation: contraction of indices

There is another useful way to understand the passage from generators to traces. Kissinger starts from a monoidal signature: a collection of types together with generating operations having finite lists of typed inputs and outputs. From this data he constructs a free abstract tensor system. Tensor expressions may be juxtaposed, relabelled, and, crucially, an upper and lower index of the same type may be contracted.

In Einstein notation a repeated upper and lower label is therefore bound. In the corresponding string diagram the two occurrences are joined by a wire. Thus index contraction is the algebraic counterpart of closing compatible pieces of boundary.

Kissinger ([7]) proves that the symmetric monoidal category associated to the free abstract tensor system is the free traced symmetric monoidal category on the original monoidal signature. We may therefore regard

$$\text{typed generators} \longrightarrow \text{tensor expressions} \longrightarrow \text{contractions}$$

as an algebraic presentation of the same traced process syntax which Spivak–Schultz–Rupel describe geometrically by oriented 1-cobordisms.

For a general Σ -net one needs the richer symmetric data carried by its graphical-species presentation, so we should not identify this construction outright with Raynor’s free wheeled prop. But it gives a particularly transparent local model of what a wheel does: a matching output and input become a bound internal index.

4 Doi–Peliti semantics and internal circulation

The free wheeled prop provides the process syntax. Doi–Peliti theory supplies a particularly suggestive operator semantics for stochastic Petri nets and reaction networks. The two should be related, but not identified.

4.1 A catalytic strand

Consider the reaction

$$C + A \longrightarrow C + B.$$

The species C is required for the reaction but is returned afterwards. Graphically the transition contains a through-strand

$$C \longrightarrow C$$

beside the conversion $A \rightarrow B$, so it has the form

$$f : C \otimes A \longrightarrow C \otimes B.$$

As written, the C -strand is still part of the external interface. Only after contraction do we obtain

$$\mathrm{Tr}^C(f) : A \longrightarrow B.$$

It is therefore useful to keep three stages apart:

$C + A \rightarrow C + B$	$\longrightarrow$	$f : C \otimes A \rightarrow C \otimes B$	$\longrightarrow$	$\mathrm{Tr}^C(f) : A \rightarrow B$
reaction syntax		open C -through-process		wheeled contraction.

The middle stage is important. The C -strand is present, but it is still externally visible. Only the final trace closes that strand and removes C from the boundary.

Now interpret the places as particle species. For each species X introduce creation and annihilation operators $a_X^\dagger$ and a_X . The jump part of the reaction above contains

$$a_C^\dagger a_B^\dagger a_C a_A.$$

Operators associated with different species commute, so this may be rearranged as

$$(a_C^\dagger a_C) a_B^\dagger a_A = N_C a_B^\dagger a_A, \quad N_C = a_C^\dagger a_C.$$

The open C -through-process has therefore become the number operator N_C . Acting on a state containing n_C copies of C , it supplies the multiplicity factor n_C . This is exactly the combinatorics that the whole-grain presentation keeps visible at the level of individual arcs.

If the transition requires and returns m copies of C , the corresponding factor is

$$(a_C^\dagger)^m a_C^m = N_C(N_C - 1) \cdots (N_C - m + 1).$$

Thus the whole-grain input profile and the falling-factorial combinatorics of annihilation operators fit together very naturally.

4.2 Do not identify the number operator with the trace

The equation $a_C^\dagger a_C = N_C$ is suggestive, but it is not itself the categorical trace of the C -wire. The operator is the semantics of the open through-process. Categorical trace performs the additional operation

$$f : C \otimes A \longrightarrow C \otimes B \quad \mapsto \quad \mathrm{Tr}^C(f) : A \longrightarrow B,$$

which removes the C -interface altogether. We should therefore distinguish

through-wire $C \rightarrow C$	number operator $a_C^\dagger a_C$	closed C -interface Tr^C .
-----------------------------------	--------------------------------------	--

The second is a representation of the first. The third is a further operation of hiding the boundary.

4.3 Beyond static catalysts

The most interesting wheels need not arise from something unchanged at every transition. Consider



Neither reaction contains a static catalytic C -through-wire. Yet after composition there is an internal circulation

$$C \rightsquigarrow D \rightsquigarrow C.$$

If the C/D -behaviour is hidden from the external boundary, the composite exposes only

$$A \otimes E \longrightarrow B \otimes F.$$

The circulating entity changes its form locally but is regenerated globally. This is a broader and more interesting interpretation of a wheel than a permanently unchanged catalyst. Enzymes with intermediate states, cofactors that are consumed and regenerated, and carrier molecules alternating between forms all have this flavour.

4.4 Stochastic dynamics

Suppose a transition τ has input multiplicity vector $s(\tau)$, output multiplicity vector $t(\tau)$, and rate r_τ . Writing

$$a^{s(\tau)} = \prod_i a_i^{s_i(\tau)},$$

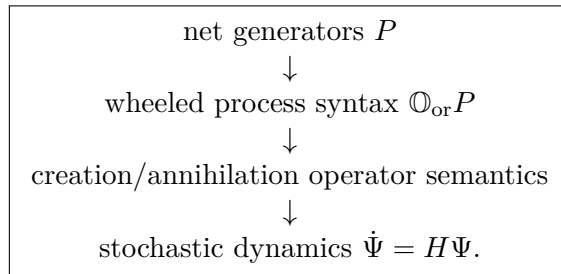
a stochastic reaction network gives rise to an operator of the familiar form

$$H = \sum_\tau r_\tau \left((a^\dagger)^{t(\tau)} - (a^\dagger)^{s(\tau)} \right) a^{s(\tau)}.$$

The first term performs the jump, while the second supplies the diagonal correction required for conservation of total probability. The probability state evolves by

$$\frac{d}{dt} \Psi(t) = H \Psi(t).$$

The point for us is the sequence of roles:



The wheeled prop supplies syntax for composition and contraction. The Doi–Peliti representation assigns operators to the process data. The master equation supplies time evolution.

The analogy with the automaton case is now much sharper. Gustafson et al. sum Booleanly over finite state histories. Doi–Peliti sums, in a far richer linear setting, over population histories. The passage from an automaton to a reaction network is therefore not a jump from one unrelated subject to another: it replaces a basis of control states by a basis of occupation-number states and replaces Boolean existence by stochastic weighting.

5 Three meanings of closing

We have now accumulated several operations which look similar diagrammatically. They should be distinguished explicitly.

Closing a systems interface

Earlier in the tutorial we wired systems together until no external signals remained. A Moore component and a Mealy component, for example, could be connected to form a closed-loop system. This is closure by *wiring components*.

Closing a wire in process syntax

In a wheeled prop we may take

$$f : X \otimes U \longrightarrow Y \otimes U$$

and contract the U -boundary to obtain

$$\text{Tr}^U(f) : X \longrightarrow Y.$$

This is closure by *graphical contraction*.

Taking a trace after representation

After applying some semantic functor

$$F : \mathcal{P} \longrightarrow \mathcal{V},$$

we may encounter a genuinely algebraic trace, state sum, integral, partition function or contraction of indices. This is closure by *evaluation*. In the Gustafson et al. example these levels happen to line up beautifully: gluing the interval to a circle is sent by the Boolean TQFT to the algebraic

trace of the transition operator. That compatibility is a theorem of the chosen semantics, not a definition of graphical trace in general.

Thus we should keep the warning

systems feedback $\neq$ wheel in process syntax $\neq$ operator trace in a chosen semantics.
--

The interesting mathematics lies in asking when these different operations are compatible.

5.1 Representing the free wheeled prop

Suppose P is a Σ -net and $\mathbb{O}_{\text{or}}P$ its free wheeled prop. One may seek a trace-preserving symmetric monoidal functor

$$F : \mathbb{O}_{\text{or}}P \longrightarrow \mathcal{V}$$

into some semantic category $\mathcal{V}$. Different targets give very different interpretations: relational or Boolean, linear, stochastic, Doi–Peliti/Fock-space, or TQFT-like. The same wheeled syntax can therefore support many valuation semantics.

The automaton paper gives the cleanest model of what we would like. In the $(1, 1)$ -generated case, the geometric closing of a defect-decorated interval becomes the algebraic trace of the corresponding endomorphism. For a general Petri-net-generated wheeled prop the analogous statement is a research question: what should a representation do with arbitrary wheeled contraction, especially in the Doi–Peliti setting?

5.2 From cobordisms to wheeled graphs

We can now state the relation among the three main pieces of the chapter more sharply. Spivak–Schultz–Rupel identify traced string-diagram syntax with labelled oriented 1-cobordisms. Gustafson et al. evaluate the $(1, 1)$ -generated fragment on intervals and circles. Raynor’s framework enlarges the syntax from one-dimensional strands to arbitrary directed graphs, with wheeled contraction playing the role of closing matching graphical interfaces.

Thus for a Petri net P we retain the semantic chain

$$P \longrightarrow \mathbb{O}_{\text{or}}P \xrightarrow{F} \mathcal{V}.$$

The first arrow freely generates composite and wheeled processes; the second assigns values to them. In the automaton corner the closing operation has both a cobordism presentation and a Prof-shadow interpretation. For a general Petri-net-generated wheeled prop, one of our main questions is how much of this compatibility survives, especially under Doi–Peliti representation.

6 Where do runners enter?

We should now return to the central architecture of the tutorial. A runner does not merely assign a number to a process. It says how one theory is *realised* using another theory together with hidden state. In the ordinary Set-based setting we wrote

$$\theta_X : T(X) \times S \longrightarrow R(X \times S).$$

The state S persists as operations are executed. That is a different job from a functor

$$F : \mathbb{O}_{\text{or}} P \longrightarrow \mathcal{V}$$

which evaluates process diagrams.

Thus there are two semantic directions:

runner semantics How is this process implemented?	valuation semantics What value does this process have?
--	---

These should not be merged. Instead we should ask how they interact.

The whole discussion can be summarised by four roles:

$$\boxed{\text{generation} \longrightarrow \text{running} \longrightarrow \text{closure} \longrightarrow \text{valuation}.}$$

Generation constructs composite process syntax from primitive transitions. Running supplies an implementation with persistent state. Closure hides compatible internal interfaces. Valuation sends the resulting process to a semantic object: a Boolean value, relation, linear map, stochastic operator, partition function, or something else.

6.1 A compatibility question for Run

Suppose a process theory T admits a wheeled or traced completion $\text{Wheel}(T)$, and suppose we have a runner

$$T \rightsquigarrow R.$$

Under what conditions does the runner extend to

$$\text{Wheel}(T) \rightsquigarrow \text{Wheel}(R)?$$

Informally,

$$\boxed{\text{does implementation commute with hiding an internal loop?}}$$

This should not be assumed automatically. A runner may expose state which is invisible at the abstract level, so closing an interface before implementation and closing it afterwards may require additional compatibility conditions. But this is precisely the sort of question the double-categorical organisation of runners is designed to make askable.

There is a useful echo here of the automaton example. Passing from interval behaviour to circular behaviour changes which internal state histories are admitted by the observational context. In a runner setting the same issue becomes richer: a hidden loop may include both the state of the abstract process and the persistent state carried by its implementation.

6.2 Should Beh preserve the shadow?

The Prof viewpoint gives a sharper formulation of this compatibility question. Suppose

$$\text{Beh} : \mathbf{Run}(\mathcal{D}) \longrightarrow \mathbf{Prof}$$

is the behavioural double functor, and let an endo-runner on T be sent to an endoprofunctor

$$P : \mathcal{B}_T \rightharpoonup \mathcal{B}_T.$$

Then **Prof** already has a canonical way to close that horizontal behaviour:

$$\langle P \rangle = \int^{x \in \mathcal{B}_T} P(x, x).$$

This suggests that the right structural question is not simply whether **Run**($\mathcal{D}$) itself is traced. It may instead be whether the particular closing operations available on suitable runners are carried by Beh to the shadow in **Prof**. Schematically, one would like a compatibility of the form

$$\begin{array}{ccc} \text{endo-runner} & \xrightarrow{\text{Beh}} & \text{endoprofunctor} \\ \downarrow \text{closure} & & \downarrow \text{shadow} \\ \text{closed run} & \longrightarrow & \text{cyclic behaviour.} \end{array}$$

Whether this can be formulated as a shadow-preserving double functor, a morphism of shadows on a suitable sub-double-category, or only as a comparison map remains to be established. The distinction is important: we do not want to manufacture a trace on all of **Run**($\mathcal{D}$) merely because the target **Prof** has a shadow.

The automaton case is the test example. There the behaviour category already contains based loops, the Prof shadow forgets their chosen cut, and the Boolean TQFT asks whether the resulting cyclic behaviour exists. A successful general theory should recover this case while allowing the hidden state carried by a nontrivial runner to participate in the coend.

6.3 A regulator as an internal cycle

Suppose a plant has an internal variable U , and a regulator continually reads, modifies and restores information associated with it. At the open level one may have

$$X \otimes U \longrightarrow Y \otimes U.$$

Externally the U -channel is not visible, so the regulated system behaves as

$$X \longrightarrow Y.$$

The hidden feedback channel is structurally analogous to the catalytic strand considered above. This does not mean that the regulator leaves U literally fixed; it may transform U through a complicated internal cycle. What matters is that the hidden part of the system closes coherently.

7 Enriched shadows and resource-valued closed behaviour

The discussion so far has mostly used ordinary profunctors. But several earlier parts of the tutorial replaced **Set** by a richer symmetric monoidal base $\mathcal{V}$. In particular, Part VI used

$$\mathcal{V} = C\text{-}\mathbf{Set} = [C, \mathbf{Set}]$$

with Day convolution in order to make cost and potential intrinsic to the semantic category. The trace story has an enriched analogue.

For a suitable cocomplete symmetric monoidal base $\mathcal{V}$, the bicategory

$$\mathcal{V}\text{-}\mathbf{Prof}$$

of $\mathcal{V}$ -categories and $\mathcal{V}$ -profunctors carries the expected shadow. If

$$P : \mathcal{A} \rightharpoonup \mathcal{A}$$

is an endoprofunctor, then its shadow is the enriched coend

$$\langle P \rangle = \int^{a \in \mathcal{A}} P(a, a) \in \mathcal{V}.$$

Thus the ordinary set of cyclic behaviours is replaced by a $\mathcal{V}$ -object of cyclic behaviours. This is exactly the form one would expect from the automata example of Gustafson–Im–Kaldawy–Khovanov–Lihn. For a nondeterministic automaton, a letter may be represented by a Boolean matrix

$$M_a(q, q') \in \mathbb{B}.$$

For a word $w = a_1 \cdots a_n$, composition gives

$$M_w = M_{a_n} \cdots M_{a_1},$$

and the circular behaviour is obtained by taking the Boolean trace

$$\mathrm{tr}_{\mathbb{B}}(M_w) = \bigvee_q M_w(q, q).$$

This is 1 precisely when there exists a closed run labelled by w .

The enriched version simply replaces Boolean matrices by $\mathcal{V}$ -valued profunctors. If the state set Q is regarded as a discrete $\mathcal{V}$ -category and

$$M_a : Q \rightharpoonup Q$$

is the profunctor associated with the letter a , then

$$(M_b \odot M_a)(q, q'') = \int^{q'} M_a(q, q') \otimes M_b(q', q'')$$

is the enriched analogue of matrix multiplication. For the word w ,

$$M_w = M_{a_n} \odot \cdots \odot M_{a_1},$$

and its circular behaviour is

$$\boxed{\langle M_w \rangle = \int^{q \in Q} M_w(q, q).}$$

The Boolean loop language is therefore one special case of the shadow construction.

7.1 The cost-enriched case

Now take

$$\mathcal{V} = C\text{-}\mathbf{Set}.$$

Each hom-object

$$M_a(q, q')$$

is itself cost-indexed. Informally,

$$M_a(q, q')(c)$$

records the a -labelled transitions from q to q' available with resource profile c .

Composition uses Day convolution, so resource values combine according to the monoidal structure on C . Schematically, a composite

$$q \xrightarrow[a]{c_1} r \xrightarrow[b]{c_2} q'$$

contributes at the combined resource level determined by

$$c_1 + c_2.$$

The shadow

$$\langle M_w \rangle = \int^q M_w(q, q)$$

is therefore a C -set of closed w -behaviours. This is the resource-sensitive analogue of the Boolean loop language: instead of asking merely

does a closed run exist?

we retain

which closed runs exist, and with what resource profile?

That distinction fits the philosophy of Part VI. There, potential was not attached externally as a number beside a state; it was represented inside the monoidal category of resource-indexed objects. Here, likewise, the result of closing a run need not collapse immediately to a scalar. The shadow itself is a structured object carrying the resource information around the loop.

7.2 Shadow first, valuation second

As before, it is useful to distinguish closure from numerical evaluation. The shadow gives

$$\langle P \rangle \in \mathcal{V}.$$

A further valuation may then send this structured cyclic behaviour to some simpler quantity,

$$\nu : \mathcal{V} \longrightarrow \mathcal{W}.$$

Thus the general pattern is

$$P \longmapsto \int^a P(a, a) \longmapsto \nu \left(\int^a P(a, a) \right).$$

Different choices of $\mathcal{V}$ and ν recover familiar notions. With Boolean enrichment one obtains existence of a closed run. With tropical or ordered enrichment one may extract a least-cost closed run. With probabilistic or linear enrichment one may obtain a weighted trace or partition-function-like quantity. With C -**Set** one retains the whole cost-indexed object before making any such collapse.

This suggests an enriched version of our behavioural construction,

$$\text{Beh}_{\mathcal{V}} : \text{Run}(\mathcal{D}) \longrightarrow \mathcal{V}\text{-}\mathbf{Prof}.$$

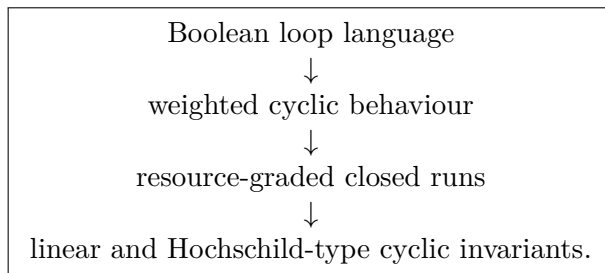
For an endo-runner $r : T \rightsquigarrow T$, the corresponding cyclic behaviour would be

$$\text{Cyc}_{\mathcal{V}}(r) = \langle \text{Beh}_{\mathcal{V}}(r) \rangle = \int^{x \in \mathcal{B}_T} \text{Beh}_{\mathcal{V}}(r)(x, x).$$

In the amortisation setting this would be a resource-indexed object of closed behaviours. One could then ask, for example, for loops in which potential is deposited and later discharged while the observable system returns to the same behavioural state. This is stronger than assigning a cost to individual transitions. It measures the resource profile of an entire closed execution.

7.3 From Boolean traces to cyclic invariants

There is a useful hierarchy here:



For the identity profunctor on a $\mathcal{V}$ -category $\mathcal{A}$,

$$\langle 1_{\mathcal{A}} \rangle = \int^a \mathcal{A}(a, a).$$

In linear settings this is the familiar HH_0 -shaped construction, and higher refinements lead naturally towards Hochschild and cyclic homology.

Thus the loop language of an automaton, the resource profile of a closed amortised run, and the linear traces appearing in field-theoretic semantics all belong to one broad pattern:

open behaviour $\longrightarrow$ shadow $\longrightarrow$ cyclic valuation.

The enriched setting therefore does not alter the trace story. It reveals that the Boolean case was only its simplest decategorified instance.

8 Refinement and layered wheeled props

There is a further reason the wheeled formulation is attractive. Suppose a high-level net P_{high} is implemented by a more detailed net P_{low} . A high-level place may be realised by a low-level marking, and a high-level transition by an entire low-level process. Such a refinement should induce a map

$$\mathbb{O}_{\text{or}}(P_{\text{high}}) \longrightarrow \mathbb{O}_{\text{or}}(P_{\text{low}}).$$

If the process theories are wheeled, refinement should preserve contraction. Thus for a high-level process f ,

$$F(\text{Tr}(f)) = \text{Tr}(F(f)).$$

This says that hiding a high-level internal cycle and then refining should agree with refining the implementation and then hiding the corresponding low-level cycle. That is exactly what one would want from a compositional notion of implementation.

This brings the theory close to the programme of layered monoidal theories ([8, 9]). In the present case each layer carries wheeled structure and the translations between layers preserve it, giving what we may call *layered wheeled props*.

8.1 Zooming in and out

Raynor's bar construction suggests a complementary picture. Repeated application of the free-process monad generates nested networks of networks. Face maps collapse levels of composition and

degeneracies insert trivial levels. One may therefore think of the bar construction as a mechanism for zooming in and out between descriptions of a process.

At one scale a complicated subnet may be treated as a single transition; at a finer scale it unfolds into a network of subprocesses. If traces are present, an apparently simple high-level feedback loop may unfold into a complicated low-level circulation. Trace preservation says that these descriptions remain compatible.

8.2 Refinement and layered wheeled props

There is a useful strengthening of the simple picture in which a whole high-level process is translated into a low-level one. Lobski and Zanasi’s original theory of layered props was designed precisely so that descriptions at several levels could coexist in a single string diagram. In particular, their language permits *partial explanations*: one region of a diagram may be translated to a finer level while the remainder is left untouched.

This is particularly natural for Petri-net refinement. A high-level transition may be replaced by a low-level subnet or process while neighbouring transitions remain at the high level. Thus one should not picture refinement only as a map

$$\mathbb{O}_{\text{or}}(P_{\text{high}}) \longrightarrow \mathbb{O}_{\text{or}}(P_{\text{low}}),$$

although such a map remains an important limiting case. More generally we want diagrams whose regions may live at different levels of description, together with explicit translations between those levels.

The 2026 development replaces the earlier PROP-specific framework by layered monoidal theories. Several monoidal theories, and translations between them, may now be combined within one string diagram. Our proposed layered wheeled props should therefore be understood as a trace-bearing specialisation of this broader framework.

The extra requirement supplied by the wheeled structure is compatibility with contraction. For a global translation F this takes the familiar form

$$F(\text{Tr}^U(f)) = \text{Tr}^{F(U)}(F(f)).$$

But the layered point of view asks for more: the same compatibility should hold when only part of the diagram inside or around the traced region is refined. We should be able to zoom into an internal implementation without changing the externally visible closed behaviour.

This is exactly the systems-theoretic reading we want. A wheel may hide a complicated internal circulation. Refinement replaces part of that circulation by a more detailed implementation; trace preservation says that hiding and refinement remain coherent.

8.3 Weighted refinement and path sums

There is also a natural probabilistic direction. A coarse transition may refine not to one lower-level process but to several possible processes, perhaps with weights. Schematically,

$$t \rightsquigarrow \sum_i p_i \rho_i.$$

One may then ask for a stochastic version of trace preservation combining weighted refinement with wheel contraction. This would fit naturally with stochastic Petri nets and with the path-sum intuition already visible in the automaton TQFT.

The hierarchy is suggestive. Boolean automata sum over possible finite state histories. Weighted automata refine this to semiring-valued path sums. Stochastic Petri nets and Doi–Peliti theory replace finite control states by occupation-number states and attach rates to population-changing events. The common question is how much of this path-sum semantics respects graphical closure.

There is also a suggestive semantic connection with the rest of this part. In their fibrational semantics, Lobski and Zanasi describe a displayed category over a base $\mathcal{X}$ by a normal lax functor

$$D : \mathcal{X} \longrightarrow \mathbf{Prof}.$$

The profunctor over a base arrow records the ways in which objects in the corresponding fibres are connected above that arrow, and composition is given by the usual profunctor coend.

This should not be identified with our proposed behavioural functor

$$\text{Beh} : \text{Run}(\mathcal{D}) \longrightarrow \mathbf{Prof},$$

but the parallel is worth noting. Both use profunctors to retain the space of mediating realisations rather than collapsing it to a function. Since $\mathbf{Prof}$ also carries a shadow, the interaction between layered refinement, behavioural semantics and cyclic closure becomes a natural question for the present programme.

9 Loops, field theory, and the remaining question

Starting from the Doi–Peliti master operator H , one studies expressions such as

$$e^{tH}.$$

Expanding the exponential produces sums over sequences of transition events; coherent-state methods lead to functional integrals; perturbative expansions organise contributions graphically, and closed internal pieces may acquire loop weights. There is therefore a family resemblance between wheels in the free process syntax and loops in a field-theoretic expansion. But they should not be identified prematurely.

The first is a combinatorial operation available before choosing a linear representation. The second appears after representing the dynamics and expanding the resulting operator or functional integral. A worthwhile research question is therefore

how does wheeled contraction in the free process theory map to contraction and loop formation in a Doi–Peliti representation?

The automaton TQFT shows what a satisfactory answer can look like in the unary case: geometrically close the interval, and semantically take the trace of the corresponding endomorphism. For a rate-decorated whole-grain Petri net P , one would like a representation

$$\mathbb{O}_{\text{or}} P \longrightarrow \text{End}(\mathcal{F}_P)$$

with a comparably precise account of contraction. Primitive transitions should recover the usual creation-annihilation expressions, sequential composition should become operator composition, and parallel composition should interact appropriately with tensor product. The difficult part is the wheel:

$$\text{Tr}^U(f) \longmapsto ?$$

One expects a relation with summing over hidden occupation numbers, index contraction, operator traces, or loop terms in generating functionals, depending on the representation. The point is not to guess one of these in advance, but to formulate and prove the compatibility theorem.

10 Relation to higher-dimensional automata

Petri nets describe concurrency, and in Part VII we saw that higher-dimensional automata make independence geometrically explicit by means of cubes. A wheeled prop records something different. A cube says that two actions can occur independently; a wheel says that some output-input dependency has been closed into an internal cycle. Thus

cubical dimension records independence,
wheeled structure records contraction and circulation.

Neither replaces the other. The automaton case again marks the one-dimensional boundary: when every transition is $(1, 1)$ -valent, connected diagrams reduce to intervals and circles; when concurrency is present, higher-dimensional cells and more general graphical contractions enter independently.

This suggests that Petri-net-to-HDA constructions may eventually be enriched by trace-sensitive information. One might ask whether a cyclic process in a wheeled prop determines a distinguished class of loops or closed higher-dimensional behaviours in the associated HDA. That question belongs to the interface between this part and the higher-dimensional geometry developed earlier.

11 What Part XI has added

We began with the modest observation that systems sometimes contain feedback loops. Spivak–Schultz–Rupel make the geometry precise: traced string diagrams are labelled oriented one-dimensional cobordisms, so closing a wire is literally a gluing operation. The automaton example then shows that even in the simplest finite-state setting this closure changes the behavioural question. A decorated interval detects ordinary accepted behaviour; closing it to a circle detects cyclic behaviour, and the Boolean TQFT evaluates that closure by a trace. The Prof shadow supplies the corresponding categorical operation on behaviour: based loops in $\mathcal{B}_T(x, x)$ pass through the coend $\int^x \mathcal{B}_T(x, x)$ to cyclic behaviour with no preferred cut.

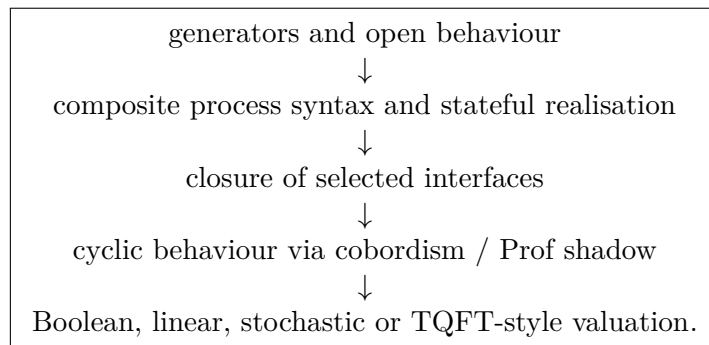
Whole-grain Petri nets generalise the same geometry from $(1, 1)$ -valent transitions to arbitrary finite input and output profiles. As oriented graphical species they freely generate wheeled process theories

$$\mathbb{O}_{\text{or}}P,$$

where sequential firing, parallel firing, identities and contraction belong to one coherent algebraic structure.

Doi–Peliti semantics then gives a particularly suggestive linear interpretation. An open catalytic through-strand $C \rightarrow C$ becomes the number operator $N_C = a_C^\dagger a_C$; more complicated internal circulations may return a carrier only after a composite cycle; and the master equation adds stochastic dynamics to the operator semantics. The automaton TQFT tells us what to hope for: a semantics in which graphical closure is matched by an appropriate algebraic trace or contraction.

The roles are now clearer:



The order should not be read as a theorem saying that every construction literally factors in this way. Part of the future mathematics is to determine which squares commute. Conceptually, however, the jobs are distinct. A runner tells us how a process is realised. A wheel tells us which part of its interface has been hidden by contraction. The shadow of **Prof** tells us how based behavioural loops become cyclic behaviours. A representation tells us how the resulting process is valued. One particularly sharp problem left by this chapter is therefore whether $\text{Beh} : \mathbf{Run}(\mathcal{D}) \rightarrow \mathbf{Prof}$ can be equipped with the appropriate compatibility with closure and shadow.

The sharper question with which to leave the chapter is therefore

when a pattern runs on matter, what new behaviour becomes visible
when part of the running is allowed to close back upon itself?

Further reading

For whole-grain Petri nets and their graphical-species formulation, see Kock’s work on whole-grain Petri nets and processes.

For the disconnected graphical framework, circuit algebras and the free wheeled-prop monad, see Raynor’s work on algebraic patterns and circuit algebras.

The equivalence between circuit algebras and wheeled props is developed by Dancso, Halacheva and Robertson.

For the geometric presentation of trace, see Spivak, Schultz and Rupel on labelled oriented one-dimensional cobordisms. For shadows and their cyclicity in bicategories, see Ponto and Shulman; the Prof shadow used here is the coend of an endoprofunctor along its diagonal.

For Doi–Peliti methods in stochastic reaction networks, the relevant literature develops the representation of reaction generators by creation and annihilation operators and derives master equations, coherent-state representations and path-integral formulations.

Finally, layered monoidal theories provide the natural surrounding framework for refinement between descriptions at different scales; in the present setting the corresponding specialisation is to layered wheeled props.

References

- [1] A. Joyal, R. Street, and D. Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society* 119 (1996), 447–468.

- [2] K. Ponto and M. Shulman. Traces in symmetric monoidal categories. *Expositiones Mathematicae* 32(3) (2014), 248–273.
- [3] K. Ponto and M. Shulman. Duality and traces for indexed monoidal categories. *Theory and Applications of Categories* 26 (2012), 582–659.
- [4] K. Ponto and M. Shulman. Shadows and traces in bicategories. *Journal of Homotopy and Related Structures* 8 (2013), 151–200.
- [5] D. I. Spivak, P. Schultz, and D. Rupel. String diagrams for traced and compact categories are oriented 1-cobordisms. *Journal of Pure and Applied Algebra* 221(8) (2017), 2064–2110.
- [6] P. Gustafson, M. S. Im, R. Kaldawy, M. Khovanov, and Z. Lihn. Automata and one-dimensional TQFTs with defects. arXiv:2301.00700, 2023.
- [7] A. Kissinger. Abstract Tensor Systems as Monoidal Categories. arXiv:1308.3586, 2013.
- [8] L. Lobski and F. Zanasi. Layered Monoidal Theories I: Diagrammatic Algebra and Applications. 2602.19776, 2026.
- [9] L. Lobski and F. Zanasi. Layered Monoidal Theories II: Fibrational Semantics. 2602.22373, 2026.
- [10] Z. Dancso, I. Halacheva, and M. Robertson. Circuit algebras are wheeled props. *Journal of Pure and Applied Algebra* 225(12) (2021), 106767.
- [11] S. Raynor. Functorial, operadic and modular operadic combinatorics of circuit algebras. arXiv:2412.20260, 2024.
- [12] S. Raynor. Modular operads, distributive laws and a nerve theorem for circuit algebras. arXiv:2412.20262, 2024.
- [13] H. Sati and U. Schreiber. The Quantum Monadology. *Quantum Studies: Mathematics and Foundations* 12 (2025), Article 25; arXiv:2310.15735.