The Organization of Abstract Organisms

Higher topoi, operadic production, and thermodynamic viability

Organisms, viruses, and computers as bounded systems of production and repair

Anonymous review copy

Prepared for Higher Categorical Semantics

Abstract

A physical boundary determines which operations produce and restore the machinery that sustains a system. We formulate this distinction using images and effective descent in a higher topos, and give a finitary operadic semantics for recursive production. The least fixed point consists exactly of the locally realizable outputs of finite derivation trees. A coupled host contributes mixed trees through an operadic pushout, and a realization map measures the image of their physical effects. Where the operation profiles are space-valued, homotopy-coherent gluing retains the spaces of realizations and their symmetries.

Budget grading separates attained feasibility from infimum cost. Discrete energy ledgers and controlled dynamics then put feasible maintenance on a single space of states. The greatest fixed point on that space is the exact locus of repeatable maintenance. For finite state graphs, a viable state reaches a directed cycle. For cyclic schedules, the minimum battery is the largest cumulative deficit on a circular interval. Together these results provide certificates for endogenous autonomy, host-relative closure, and industrially supported maintenance. A suite of finite models compares structural dependence, threshold effects, and observational equivalence, with exact calculations of budgets, viable states, and admissible trajectories.

Keywords Autopoiesis, closure of constraints, higher topos, effective descent, colored operad, viable dynamics, exergy, virus, computation.

Source and preparation The conceptual starting point is the three-essay reading edition *Organisms, viruses, and computers*. The manuscript gives separate treatments of production images, operadic coupling, and thermodynamic viability, and connects them through typed finite witnesses. AI assistance was used in drafting, mathematical exposition, and the generation and checking of the model atlas. [1, 2, 3]

Contents

1 The boundary of an organism 1

2 Images and organizational demands 4

3 Supplementation and descent 7

4 Geometric invariance and local truth 10

5 Recursive production in a higher topos 13

6 Calculi of closure 17

7 Open processes and operadic witnesses 20

8 Coupling along interfaces 23

9 Stocks, catalysts, and regeneration 26

10 Budget grades and attainable realizations 28

11 Optimal finite proofs and portfolio costs 31

12 Exergy ledgers and finite resources 34

13 Dissipation in explicit process models 36

14 Viability as a greatest fixed point 39

15 Controllers, games, and probabilities 42

16 A common state for production and maintenance 45

17 Periodic supply and exact battery thresholds 48

18 The invariant of a bounded system 51

19 Observations and physical equivalences 54

20 A regenerative cellular model 56

21 A host-dependent replicator 58

22 Computing machinery and fabrication 60

23 Heredity, export, and repeated birth 63

24 Design and minimal supplementary capacity 66

25 Context-dependent regulation 68

26 Symmetry and coherent maintenance witnesses 71

27 Composition of viability regimes 73

28 Autocatalysis and the restoration of constraints 75

29 Temporal resources beyond periodic forcing 77

30 Coarse regimes and testable distinctions 80

31 Production conditioned by a viable cohort 83

32 Worst-case timing on a disturbance graph 86

33 Decision procedures and witness size 88

34 Common certificates for local organization 90

35 Finite verification and reproduction 92

36 A guide to the finite atlas 96

37 Periodic maintenance schedules 98

38 Typed sources and attainable repair 114

39 Possible, robust, and stochastic repair 130

40 Selectable maintenance routes 146

41 Stock and regeneration schedules 162

42 Local closure on finite context spaces 178

CHAPTER 1

The boundary of an organism

1.1 Three questions of organization

An organism preserves an organization through processes that consume, produce, and exchange material. The material of a membrane is replaceable. The role of the membrane as a selective boundary persists by means of activities distributed across the system. A description of these activities has three different questions to answer. It must specify what the system requires, which operations produce those requirements, and when those operations can recur in a viable history.

The source essays isolate these questions in a sequence. The first essay compares images of demand and endogenous supply. The second replaces a snapshot of supply with a recursively generated system of operations. The third measures feasible realization and continuing viability. We use that sequence to construct a single semantics. The image calculation retains the locality of available witnesses. The operadic calculation retains how their components are assembled. The state calculation accounts for the time and resources of their repeated execution. [1, 2, 3]

There are three kinds of boundary data throughout the paper. A *material typing* assigns types to components. A *production presentation* assigns witnesses to organizational roles. An *operating regime* specifies the transitions, costs, and environmental inputs through which the roles are maintained. A regime records resource exchanges at its boundary. A nutrient arrival is an admissible input in a regime whose machinery transforms the nutrient. A delivered replacement is a supply of a different organizational role. Keeping these two roles distinct gives a precise language for the relation between a virus and a host, and between a computer and a fabrication chain.

1.2 The life of a role

Use a for an organizational role, R_a for its possible instances, and $D_a \to R_a$ for the family of demands. Instances can include a particular interface or a specific quantity. A production map $P_a \to R_a$ records the endpoint of a witness. Its image classifies the demands with local witnesses. When P_a consists of finite derivations, the map forgets the tree while preserving existence. When a tree is realized in a physical operad, it can instead retain a space of processes above each instance. Both forms of data enter the paper.

Three operations on such a model serve different purposes. Restriction changes the context in which the same system is inspected. Coupling adds processes through an interface. Changing the organizational signature changes the demand itself. Closure may

increase when a signature is reduced, when a supplement is added, or when a context selects part of a family. The theorems distinguish these ways of obtaining an inclusion by identifying the arrows that produce them.

A familiar causal pattern illustrates the distinction. A transport process makes a feedstock available to a conversion process. The conversion process makes material needed to repair transport. A finite bootstrapping tree describes the production of an additional transport component from seeded machinery. A recurrent schedule describes the restoration of that seed. The cycle lives in the dynamics of production and consumption, and a finite tree lives in the inductive certificate for an individual output.

1.3 Mathematical organization

Chapters 2–4 give the image semantics in slices of a higher topos. They prove descent for the order between demand and supply, a comparison formula for supplements, and a contextual classification on a finite catalog. Chapters 5–9 give the least-fixed-point and operadic semantics. They construct finite certificates, identify the image of coupling by the mixed terms that realize it, and refine availability to inventory when multiplicity matters.

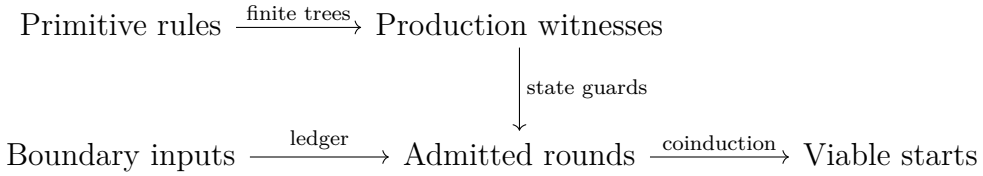
Chapters 10–13 develop the budget and physical accounting. Scalar and vector grades record the cost of particular realizations. A budget can be tested for attainment by an image, and ordered grades track how witnesses compose. Chapters 14–17 give greatest-fixed-point viability, the finite graph and game results, and exact battery tests. The key integration is the transition which checks a repair tree against the present stock, then updates that stock and the common ledger.

Chapters 18–24 compare the material, causal, and operational invariants, prove separation for explicit cell, virus, and device systems, and analyze lineage and design. The appendix contains an executable finite procedure and the atlas records exact witnesses and threshold calculations for several families. The derivations behind the tables appear in the theorem chapters and in the atlas guide.

1.4 Reading the formulas

The paper uses μ for a least fixed point and ν for a greatest fixed point. In a slice of $\mathcal{E}$, $\mathrm{Sub}(Y)$ denotes the ordered set of equivalence classes of monomorphisms into Y. A join is an image of a coproduct. A conjunction of conditions on the same state is a pullback of subobjects. The horizontal placement of an operation's inputs in a profile indicates a finite ordered list. The symmetric actions reindex that list.

A proof of availability supplies a tree. A proof of indefinite viability supplies a postfixed subobject or, in a finite graph, a path to a cycle. An inequality of grades compares realizations at a common boundary. The labels on the viable edges identify how these three certificates refer to a single execution.



The upper horizontal arrow constructs a finite derivation from primitive rules and seeds. The vertical arrow evaluates an ordered witness against stocks, obligations, and available exergy. The lower horizontal arrow supplies the changes to those stocks during a round. The rightmost construction keeps the starts of perpetually extendable paths of admitted rounds. Its input includes the edges labeled by the same witnesses computed at the top.

An analyst can begin at either end of this diagram. Beginning with a demand, one derives a family of productive trees and checks how its realizations enter the state graph. Beginning with a viable trajectory, one reads the edge labels to recover the supplied organizational roles and their provenance. The two readings agree on the operations that were executed. The image semantics records only the existence of their effects. The operational record keeps the witnesses of how and when those effects were produced.

Write B for the context and I for eligible initial states. The recurring equations are

$$\mathrm{Req}(X) \subseteq \mu\Phi_X, \qquad \mathrm{im}\Big(I \wedge \nu\Psi_X \longrightarrow B\Big) = B.$$

In the sustained condition the viability edges use the same production and budget data as Φ_X. The image on the right of the second equation expresses that an eligible viable start exists locally throughout the context.

CHAPTER 2

Images and organizational demands

2.1 Local realizability

Fix a higher topos $\mathcal{E}$ and a context object B. We work in $\mathcal{E}_{/B}$. For $f : Y \to R$ in the slice write

$$Y \overset{e_f}{\twoheadrightarrow} \mathrm{im}(f) \overset{m_f}{\hookrightarrow} R$$

for its effective epimorphism and monomorphism factorization. The effective epimorphism is characterized by the colimit of its Čech nerve. The monomorphism is a (-1)-truncated map. These factorizations and their stability belong to the basic theory of higher topoi. [4, Ch. 6][6]

Definition 2.1. A demand diagram over B is a pair $D \xrightarrow{d} R \xleftarrow{p} P$ in the slice. Its *closure judgment* is $\mathrm{im}(d) \le \mathrm{im}(p)$. Its object of matches is $W = D \times_R P$.

Theorem 2.2 (Local witness theorem)**.** *A demand diagram is closed exactly when $W \to D$ is an effective epimorphism. The image of $W \to D$ is $d^* \mathrm{im}(p)$.*

Proof. Factor p, then pull both factors back along d. The pulled-back first factor is an effective epimorphism by universality of colimits, and the second is a monomorphism. The image is therefore $d^* \mathrm{im}(p)$. That subobject is all of D exactly when d factors through $\mathrm{im}(p)$. By the universal property of the image this is equivalent to $\mathrm{im}(d) \le \mathrm{im}(p)$. □

For a test object $T \to D$, the object $T \times_D W$ presents local realizers of its family of demands. The theorem is expressed directly by that projection. In particular, its fiber at an individual demand can retain the production method, the chosen resources, the space of admissible executions, and their automorphisms.

Example 2.3 (A torsor of implementations)**.** Work over a circle with the locally constant sheaf P of sheets of the degree-two cover, and take $R = D = 1$ in the sheaf topos. The cover is locally split and $P \to 1$ is an effective epimorphism. Monodromy interchanges the two sheets, so the space of global sections is empty. The diagram is closed by Theorem 2.2. The locus of demand satisfaction is the entire circle, with local witness choices interchanged around its generator.

2.2 The classifier of contexts

For $g : R \to B$, the functor $g^* : \mathrm{Sub}(B) \to \mathrm{Sub}(R)$ has right adjoint $\forall_g$. Write $A \Rightarrow C$ for Heyting implication in $\mathrm{Sub}(R)$. Define

$$\mathcal{L}_B(D, P) = \forall_g\big(\mathrm{im}(d) \Rightarrow \mathrm{im}(p)\big) \quad \in \mathrm{Sub}(B).$$

Proposition 2.4 (The closure locus)**.** *For $u : T \to B$ the restricted diagram is closed exactly when u factors through $\mathcal{L}_B(D, P)$. In particular, B is the closure locus of every closed diagram over B.*

Proof. Put $S = \mathrm{im}(u)$. The restriction to T has images pulled back from R. Factor u through S. The effective epimorphism $R \times_B T \to R \times_B S$ reflects subobject order. To see reflection, take any subobject M with pullback below the pullback of N. The pullback of M covers M, and its map to the ambient object factors through N. Taking its image puts M below N. It follows that restricted closure is equivalent to

$$g^* S \wedge \mathrm{im}(d) \leq \mathrm{im}(p).$$

The adjunctions for implication and universal quantification rewrite this as $S \leq \mathcal{L}_B(D, P)$. □

A fixed collection of modes Σ is handled by the intersection of its closure loci in $\mathrm{Sub}(B)$. Restriction of a mode family along an inclusion $J \subseteq \Sigma$ sends the intersection for Σ into the intersection for J. This follows from the universal property of the intersection. A dependent family can instead be packaged as a single diagram over the total object of modes and then quantified down to B.

2.3 What a certificate contains

Let $P \to R$ be a family of produced instances, let $A = \mathrm{im}(d)$, and let $P' \to P$ carry a refinement of witnesses. Composition gives $P' \to R$ and an inclusion of images into $\mathrm{im}(p)$. The inclusion is equality exactly when $P' \to \mathrm{im}(p)$ is an effective epimorphism. In one direction this is the image criterion for an epimorphism. In the other direction the effective epimorphism followed by the existing monomorphism is the image factorization.

A verification procedure may therefore store a compact family of witness representatives. It is complete when the refined witness family still covers the production image. Completeness can be checked profile by profile and then glued over the context by effective descent. The witness representatives need no globally coherent choice. The only comparison is the image of their evaluation.

Proposition 2.5 (Composition of local witnesses)**.** *Suppose $U \to D$ is an effective epimorphism and $V \to U$ is an effective epimorphism. A map $V \to P$ over $D \to R$ supplies a closure certificate for $D \xrightarrow{d} R \xleftarrow{p} P$.*

Proof. The composite $V \to D$ is an effective epimorphism, as its image is all of D. The matching map $V \to D \times_R P$ factors that composite. Taking images shows that $D \times_R P \to D$ has image D. Apply Theorem 2.2. □

2.4 Typed support

A material catalog is an object K over B. For a component object C_X and typing $\tau_X : C_X \to K$, define $\mathrm{Supp}(X) = \mathrm{im}(\tau_X)$. The mapping space of factorizations through a fixed subobject is either empty or contractible. Homotopic maps and equivalences of component diagrams consequently yield the same support. The same argument applies to the demand and production images.

Proposition 2.6 (Independent coordinates)**.** *In the category of material typings and demand diagrams, arbitrary pairs of support and closure truth occur over a context. More precisely, for $S \hookrightarrow K$ and $L \hookrightarrow B$ there is a system with $\mathrm{Supp}(X) = S$ and $\mathcal{L}_B(X) = L$.*

Proof. Take $C_X = S$ with its given typing, and in the slice choose $R = D = B$ with demand the identity. Take $p : L \hookrightarrow B$. Its image is L. Implication from the top subobject is the identity, and quantification along $B \xrightarrow{\mathrm{id}} B$ is the identity. Thus the formula for $\mathcal{L}_B$ gives L. □

Fixing S and varying L realizes distinct organizations at a common support. Fixing L and varying S realizes a common closure judgment across material supports. This elementary construction is the reference case for the stronger examples with dynamics in Chapter 18.

CHAPTER 3

Supplementation and descent

3.1 The geometry of combined supply

Let A, E, H denote the demand, endogenous supply, and supplementary supply subobjects of R respectively. The letters are used for the corresponding images when a presentation is given. Define the residual task of a proposed supplement by the inequality $A \leq E \vee H$. The join can be formed using the coproduct of presentations.

Lemma 3.1 (Image of joint supply)**.** *For maps $p : P \to R$ and $h : Q \to R$, one has* $\mathrm{im}([p, h]) = \mathrm{im}(p) \vee \mathrm{im}(h)$.

Proof. A subobject receives a factorization of the copairing exactly when it receives a factorization of each summand. The image of the copairing and the join therefore have the same upper bounds in $\mathrm{Sub}(R)$. □

A supplement is a cover of the unmet demands in the following exact sense. Pull E and H back to A. Their join is all of A exactly when the copairing of these pullbacks covers A. The match theorem refines each of those two local alternatives to concrete witnesses. This yields a two-stage certificate consisting of a cover of A by sources and a local realization above each source.

3.2 Residuals and minimal supplements

For sets, the residual $A \setminus E$ is the least possible H. A complemented E in any topos admits the same calculation with $A \wedge \neg E$. In a general Heyting lattice we describe the full upper set of possible supplements.

Theorem 3.2 (Supplement interval)**.** *Let $A, E \in \mathrm{Sub}(R)$ and suppose $E \vee F = \top$ and $E \wedge F = \bot$. The subobjects H with $A \leq E \vee H$ form the interval $[A \wedge F, \top]$. For arbitrary E, the restriction $H \mapsto H \wedge A$ preserves satisfaction of the supplement inequality in both directions.*

Proof. If $A \leq E \vee H$, distributivity gives $A \wedge F \leq (E \wedge F) \vee (H \wedge F) \leq H$. Conversely $A = (A \wedge E) \vee (A \wedge F)$, which gives sufficiency. For the restriction claim, intersect the supplement inequality with A. It becomes $A = (A \wedge E) \vee (A \wedge H)$, which implies $A \leq E \vee (H \wedge A)$. Monotonicity gives the reverse implication. □

Example 3.3 (A family with no smallest supplement)**.** In the sheaf topos over $\mathbb{R}$, use $R = A = 1$ and let E be the open $\mathbb{R} \setminus \{0\}$. A supplement is exactly an open neighborhood of 0. The intersection in the lattice of opens of the intervals $(-1/n, 1/n)$, $n \geq 1$, is the

interior of $\{0\}$, hence empty. Each interval supplies a supplement and their meet lies outside the upper set of supplements. The upper set has no least element.

A finite collection of available supplements determines a separate combinatorial problem. If $H_1, \ldots, H_n$ are their supply images and the demands are a finite set, define $I(a) = \{i \mid a \in H_i\}$ for each $a \in A \setminus E$. A portfolio $J \subseteq [n]$ supplies all demands exactly when $J \cap I(a)$ is inhabited for every such a. Thus the minimal portfolios are the minimal hitting sets of the family $I(a)$. This result follows by elementwise membership in the finite union. It distinguishes a least subobject of missing resources from a catalog of ways to cover those resources.

3.3 Descent and stability

Theorem 3.4 (Image descent)**.** *Let $u : U \twoheadrightarrow B$ be an effective epimorphism. A demand diagram is closed over B exactly when its restriction to U is closed. The same statement holds for any supplement diagram using the combined production map.*

Proof. Images commute with slice pullback. For reflection, apply the argument in Proposition 2.4 to the effective epimorphism $R \times_B U \to R$. For a supplement, use the combined map $[p, h]$. Pullback commutes with that coproduct because colimits are universal in $\mathcal{E}$. □

The source distinction between observation and contribution now has an invariant formulation. An effective cover preserves the closure judgment in both directions. Adding a supplement changes the image by a join and can produce strict enlargement. For example, over a point take $R = \{a, b\}$, $A = R$, $E = \{a\}$, $H = \{b\}$. The joint image is all of R, and every effective cover preserves the properness of the endogenous image.

Proposition 3.5 (Two-step gluing of requirements)**.** *For $D \simeq \operatorname{colim}_i D_i$ over R, the diagram $D \to R \leftarrow P$ is closed exactly when every $D_i \to R \leftarrow P$ is closed.*

Proof. For any subobject $M \hookrightarrow R$, the space of maps from D to M over R is the limit of those from D_i. These spaces are propositions. Their limit is inhabited exactly when all are inhabited. Apply this with $M = \operatorname{im}(p)$. □

3.4 Three supplementation levels

A model can track feedstock, machinery, and execution separately. Let A^{feed} denote admissible environmental inputs. These live in the source interfaces for the production rules. Let A^{mach} be organizational demands. These live in the target of the production witness map. Let A^{state} be a constraint on states where the witnesses can operate. These live in a different object, the state space of a transition system. Importing a source element can make a rule executable, delivering a machine can add a target witness, and adding power can enable a transition. The three changes are implemented by three different diagrams.

For example, a transport certificate $e : (f, m) \to p$ consumes a feedstock f using machinery m to produce precursor p. Supplying f changes the leaf data. Supplying m changes an organizational input. Increasing a battery to make e executable changes a guard on a realization of the same operation. A proof of closure must still produce each organizational target. A proof of viability must execute the appropriate proofs in states where all guards hold.

3.5 Recording the point of entry

A supplement certificate can report the first internal operation that uses an imported resource. To give such a report in a rule presentation, label the imported leaf and follow the path from that leaf to the root. The first rule on that path is the boundary-facing use of the resource. Two distinct imported leaves can have the same first use when they enter one assembly, or distinct first uses when they enter separate branches. The full witness contains the complete assignment of leaves to uses. The source set of the tree is its projection to the names of the active suppliers.

For the transport rule e above, a feedstock certificate ends at the f input of e, while an externally fabricated machine ends at the m input. Both paths can reach p. The input typing proves that they have different organizational roles even when they reach the same production image. This is the additional provenance used to compare a feedstock boundary with a machinery boundary.

CHAPTER 4

Geometric invariance and local truth

4.1 Change of ambient topos

For a geometric morphism $f : \mathsf{F} \to \mathcal{E}$, its inverse image $f^* : \mathcal{E} \to \mathsf{F}$ preserves finite limits and small colimits. It preserves monomorphisms because it preserves diagonals and equivalences. It preserves effective epimorphisms because it preserves the Čech nerve and its colimit. Consequently it preserves images, any specified small join, and finite meets.

Theorem 4.1 (Positive invariance)**.** *Inverse image preserves closure of every demand diagram, preserves joint supply, and carries a local witness certificate to a local witness certificate. A geometric inverse image conservative on equivalences reflects closure.*

Proof. The matching object is a pullback. Preservation of effective epimorphisms proves preservation of the judgment by Theorem 2.2, and preservation of joins gives the joint-supply claim. For reflection, express $A \leq E$ as equivalence of $A \wedge E \to A$. Its inverse image is the corresponding comparison in F. Conservativity reflects that equivalence. □

The theorem is about the inclusion at a given context. The contextual locus itself is computed using a universal quantifier and implication. Both commute with base change between slices of a fixed topos. A comparison along an arbitrary geometric morphism always has the direction

$$f^*\mathcal{L}_B(D, P) \;\leq\; \mathcal{L}_{f^*B}\big(f^*D, f^*P\big).$$

Indeed the original locus satisfies the judgment after restriction. Its inverse image satisfies it by Theorem 4.1, so Proposition 2.4 gives the displayed inequality.

Example 4.2 (Strict comparison of loci)**.** Use sheaves on $\mathbb{R}$, let $R = B = 1$, take D to be the open $\mathbb{R} \setminus \{0\}$, and take $P = 0$. The closure locus is the interior of $\{0\}$, hence 0. Under the stalk at 0 the demand is empty and the produced image is empty. The target locus is the terminal set. The displayed comparison has distinct endpoints.

4.2 Positive certificates for separation

Write $\mathcal{W} = A \wedge \neg E$ for the subobject of demanded instances disjoint from endogenous supply. A witness $T \to \mathcal{W}$ proves, over T, a localized shortage. With decidable membership in E, the equality

$$A = (A \wedge E) \vee \mathcal{W}$$

also classifies every demand locally into one of the two cases. The equation follows by intersecting A with $E \vee \neg E = \top$. For the dense-open example of Chapter 3, E is proper

and $\neg E = \bot$. The order witness $A \not\leq E$ and a point of $\mathcal{W}$ encode different amounts of information.

For a finite constant catalog with explicit decidable flags, each Boolean state becomes a finite coproduct component. All formulas built using the finite truth tables for these flags are geometric combinations of the corresponding summands. This is a useful way to express an invariant that can both assert supply and certify an exact deficiency in the internal model.

Theorem 4.3 (Finite contextual strata)**.** *Let I be a finite set and let $C_i \hookrightarrow B$ be complemented subobjects. Put $C_i^1 = C_i$ and $C_i^0 = \neg C_i$. The subobjects*

$$B_\sigma = \bigwedge_{i \in I} C_i^{\sigma(i)}, \qquad \sigma : I \to \{0, 1\},$$

form a disjoint coproduct decomposition of B. Every geometric inverse image carries it to the corresponding decomposition for $f^* C_i$.

Proof. Distinct bit strings disagree at an index where the two factors are disjoint. Finite distributivity gives $\bigvee_\sigma B_\sigma = \bigwedge_i (C_i^0 \vee C_i^1) = B$. Disjoint coproducts in a topos identify the copairing of the B_σ with that join. Inverse image preserves finite intersections, coproducts, and disjointness, so it transports the complemented pairs and their products. □

4.3 Two notions of inhabitation

For a state object $Q \to B$, the judgment that every context locally contains a state is the equation $\mathrm{im}(Q \to B) = B$. For a test $T \to B$, a global choice is a section of $T \times_B Q \to T$. The latter implies the former over T. The distinction matters for viable kernels. A kernel whose fibers are two viable states can have the same monodromy as Example 2.3. Each state can carry a loop, and viable-state choices can still be exchanged by descent.

Proposition 4.4 (Viability labels on a torsor)**.** *Let $Q \twoheadrightarrow B$ be a locally constant torsor, and use the transition graph with state Q and the identity loop at every state. Its viable-state image is B. Its global viable-state choices are exactly the global sections of the torsor.*

Proof. The entire state object has the self-transition, hence is a postfixed subobject for the safe predecessor with safety region Q. The greatest such subobject is Q. The image and section assertions follow from its projection to B. □

4.4 Transport of a full record

An equivalence of bounded presentations includes an equivalence of ambient resource objects, of the demand and witness objects over them, and of the state and transition objects. It induces an order isomorphism on the relevant subobject lattices. If $\theta : L \to M$ is an order isomorphism and F, G are monotone with $\theta F = G\theta$, then

$$\theta(\mu F) = \mu G, \qquad \theta(\nu F) = \nu G.$$

The first equation follows by transporting the defining comparison with each prefixed point. The second follows by transporting the comparison with each postfixed point. This argument will be used for productions, viability, and their joint finite presentations. It preserves the value of an invariant through the underlying structure that computes it.

4.5 Certified equivalence in a chosen context

A comparison of two systems over B can be restricted to a subobject $L \hookrightarrow B$ before transporting the invariant. When the restricted records are equivalent over L, every matched demand judgment agrees over each test $T \to L$. To prove this, pull the equivalence back to T, then transport the demand and witness images. Thus the coincidence of closure is a property of the whole restricted family. At the same time the transport proof identifies the witness objects over T, so automorphisms and local choices of implementation remain part of the comparison.

If such equivalences themselves form a family with descent data on an effective cover of L, their arrows and compatibility homotopies glue to an equivalence over L. Check this levelwise on the defining diagrams. Descent for objects in slices glues their maps, and pullback along the cover reflects equivalences. The gluing criterion concerns equivalences of the diagrams. Separate local truth certificates for closure need only cover the demand, as in Proposition 2.5. The distinction will be used when production proofs carry symmetry.

CHAPTER 5

Recursive production in a higher topos

5.1 Typed witnesses

Let $K \to B$ be an object of facts. A fact is the availability of a specified capability at a specified interface. We use *fact* for the label, so that a fact carries a proposition of availability at each stage. A rule presentation consists of a seed map $S \to K$, and for each $n \in \mathbb{N}$ a family of rules E_n with n inputs and one output,

$$i_n : E_n \longrightarrow K^n, \qquad o_n : E_n \longrightarrow K.$$

Products are in the slice over B, so $K^0 = B$. A family E_n can have several realizations of a common rule, and paths in it can express changes of realization that preserve the rule. A nullary rule is a process with an empty prerequisite list. A seed is an explicit certificate available at stage zero. Both enter the least fixed point. Their costs and provenance can be assigned separately.

For $A \hookrightarrow K$, pull back $A^n \hookrightarrow K^n$ along i_n to obtain $E_n[A]$. With $S_0 = \mathrm{im}(S \to K)$, define

$$\Phi(A) = A \ \vee \ S_0 \ \vee \ \bigvee_{n \geq 0} \mathrm{im}\Big(E_n[A] \xrightarrow{o_n} K\Big).$$

The output factor ensures that a derived fact is tracked as a subobject of K. The A term records that availability proofs may be reused. An inventory semantics for consumed instances appears in Chapter 9.

5.2 Continuity and the least fixed point

Lemma 5.1 (Finite premises and directed joins)**.** *For a nonempty directed family $(A_j)_j$ in $\mathrm{Sub}(K)$, one has $\Phi(\bigvee_j A_j) = \bigvee_j \Phi(A_j)$.*

Proof. Write $A = \bigvee_j A_j$. In the product over B, A^n is covered by products of the A_{j_k} because pullback distributes over their joins one coordinate at a time. Each product of finitely many entries maps into A_j^n for an upper bound j of those indices. Hence $A^n = \bigvee_j A_j^n$ as subobjects of K^n. For $n = 0$ both sides are the terminal subobject because the index is nonempty. Pullback along i_n preserves joins. The direct image on subobjects along o_n, given by taking an image, preserves joins by its adjunction to pullback. Distributing the indicated joins in the formula for Φ gives the result. □

Theorem 5.2 (Finitary fixed point)**.** *Define $A_0 = S_0$, $A_{j+1} = \Phi(A_j)$. The join $G = \bigvee_{j<\omega} A_j$ is the least fixed point of Φ. It is the smallest subobject containing the seeds and closed under every rule.*

Proof. The stages form an ascending chain. By Lemma 5.1, $\Phi(G) = \bigvee_j \Phi(A_j) = \bigvee_j A_{j+1} = G$. Every rule-closed subobject containing S_0 contains A_0, and containment of A_j gives containment of $\Phi(A_j)$ by monotonicity. It therefore contains G. Conversely any fixed point contains S_0 and the output of every rule whose input lies in it. □

For each context, membership in A_j means derivability by a proof of height at most j, with the seed height taken to be 0. The height is a bound on nested applications, rather than a count of total resources. A large parallel assembly can have small height and high cost. Conversely, a sequence of cheap operations can have large height. This difference is used in the cost recurrence in Chapter 11.

5.3 The space of finite trees

Define $T_0 = S$ over K. Build the space of trees of height at most $j + 1$ by

$$T_{j+1} = S \amalg \coprod_{n\geq 0} \left(E_n \times_{K^n} T_j^n\right) \quad \longrightarrow K.$$

In each rule summand the root output is o_n. The maps $T_j \to T_{j+1}$ start with the seed inclusion and then apply the preceding maps to each branch. A nullary operation contributes one root and no branches. The pullbacks are homotopy pullbacks in the slice. Thus equality of a branch output with the required input is realized by a path in K. Internal symmetries can act on the tree space by transporting these matching paths.

Theorem 5.3 (Finite tree completeness)**.** *The image of $T_j \to K$ is A_j, and the image of $\operatorname{colim}_{j<\omega} T_j \to K$ is G. In particular $d : D \to K$ is covered by G exactly when finite derivations of demands exist locally along an effective epimorphism over D.*

Proof. At stage zero the equality is the definition of A_0. If the image of T_j is A_j, the map $T_j^n \to A_j^n$ is an effective epimorphism for every n, since effective epimorphisms are stable under pullback and compose. Pulling back along i_n gives an effective epimorphism from the n-ary tree summand onto $E_n[A_j]$. Therefore the two output images agree. Taking the join with S_0 gives $\operatorname{im}(T_{j+1})$. It also contains $\operatorname{im}(T_j)$ by the map between the stages, giving A_{j+1}. Images of the colimit agree with the join of the images by Proposition 3.5. The local-witness equivalence follows by pulling back this colimit's map to K along d. □

Corollary 5.4 (Geometric transport of derivations)**.** *The finite-stage tree construction, its image, and the generated subobject G commute with geometric inverse image. The closure judgment for the generated subobject descends along effective epimorphisms.*

Proof. Each stage uses finite limits, images, and coproducts. Inverse image preserves all of these and the final sequential colimit. For descent, apply Theorem 3.4 to the demand and the colimit of tree witnesses. □

5.4 Maps of presentations

A morphism of presentations over a fixed K comprises maps on seeds and rule families that preserve each input and output map up to coherent homotopy. Such a morphism evaluates every source tree as a target tree. The maps on trees are constructed by induction using the universal property of pullback. They induce $G_{\text{source}} \leq G_{\text{target}}$. An inclusion of rule catalogs is the case in which the maps on families are monomorphisms. A family of multiple process realizations can also map onto a common rule label.

A comparison for translation of facts uses an arbitrary map $k : K \to L$. Suppose the images under k of all seeds and rules are represented in a presentation on L, and the representing maps respect k^n and k. Then $\mathrm{im}(k|_{G_K}) \leq G_L$. One way to prove this is to evaluate the source tree stage by stage in the target and use the effective epimorphism of the source tree family onto G_K. This argument gives soundness of compiling one resource vocabulary into another.

5.5 A direct finite bound

Proposition 5.5 (Finite ranks)**.** *For N distinct facts in* **Set***, the chain from S_0 stabilizes after at most $N - |S_0|$ strict stages. Every available fact has a tree of height at most $N - |S_0|$. For a finite required subset D, a simultaneous height bound is the maximum of its members' first appearance stages, with the empty maximum taken as* 0*.*

Proof. Every strict step increases cardinality by at least one. A step with equality is followed by equality at every later step, since Φ is a function. The height statement is Theorem 5.3 in sets. For simultaneous requirements, the chain contains D at stage j exactly when every member's first stage is at most j. □

Consider facts b, t, f, p and a seed b. The rules $b \to t,\ (b, t) \to f,\ (f, t) \to p$ derive p at stage 3, with t at 1 and f at 2. A rule $(p, f) \to b$ witnesses reconstruction of the seed with height 4. First appearance of b is still 0. The proof that b is initially available and the proof that it can be reconstructed have different roots. We retain this distinction by reserving a separate fact label for a completed repair.

5.6 Typed provenance

Attach a source label to each seed and primitive rule. Let J be a finite set of available sources and let $\mathrm{src}(t) \subseteq J$ be the set of labels in t. The operation of joining branch trees unions their source sets and adds the label of the root. The subfamily of trees with $\mathrm{src}(t) \subseteq I$ for $I \subseteq J$ is exactly the family built from the restricted seeds and rules. This

assertion follows by induction on height, because membership in I is checked at each vertex.

If a fact has several source certificates, the inclusion-minimal source sets suffice for all portfolio queries. For a given portfolio I, availability holds exactly when I contains one of those minimal sets. Indeed the tree condition just proved gives necessity for some subset of I, and finiteness of J gives a minimal such subset. This removes duplicated sources from the logical certificate while retaining all possible enabling combinations.

5.7 Joint outputs and projections

A rule can produce a finite tuple of outputs. For a collection of Boolean facts, add one rule with the same input list for each coordinate of the tuple. After one application, the produced subobject contains exactly the union of the projections of the joint outputs. To verify this, apply the direct image along each coordinate to the object of admitted tuple realizations, then join the resulting subobjects. This gives the same update as the family of single-coordinate rules.

A demand for a joint output uses the tuple itself as a fact in a larger catalog. Separate coordinate witnesses can be chosen by a product of covers, while the joint witness has a common operation at its root. For example, two separate suppliers can both produce one of two named replacement components. A single plant furnishing both within a shared process supplies a witness in a different profile. Its two projections recover the separately available components. A common witness exists over a context exactly when the coordinate data admit the joint profile there.

CHAPTER 6

Calculi of closure

6.1 Closure with variable seeds

Fix a rule family E on a set K. Allow $S \subseteq K$ to vary and write $\gamma(S)$ for its generated set. Then γ is an extensive, monotone, idempotent map. Extensivity is the seed inclusion. Monotonicity follows by regarding each tree over S as a tree over any larger seed. Idempotence follows since $\gamma(S)$ is already closed under the rules. The same proof applies to $\mathrm{Sub}(K)$ in a higher topos, using subobjects as seed maps. These three laws make γ a closure operator on the ordered set of seeds.

Proposition 6.1 (Lattice of organizations)**.** *The closed seed sets form a complete lattice. Their meet is intersection and their join is γ applied to their union. The map γ is left adjoint to the inclusion of the closed sets.*

Proof. If C_i are closed and the premises of a rule lie in their intersection, the output lies in each C_i, hence in the intersection. The union generates the least closed upper bound by Theorem 5.2. The adjunction is the equivalence $\gamma(S) \subseteq C \;\Leftrightarrow\; S \subseteq C$ for closed C. In the subobject formulation, each rule takes inputs in the meet only when the inputs factor through every member, so the same calculation uses the order of subobjects. □

For finite K, an implication $L \rightsquigarrow k$ between a finite set of inputs L and a conclusion k is valid exactly when $k \in \gamma(L)$. The valid implications are closed under weakening of the input set and under cut. To verify cut, take $k \in \gamma(M)$ and $M \subseteq \gamma(L)$. By monotonicity and idempotence, $k \in \gamma(\gamma(L)) = \gamma(L)$. This is the composition law for finite derivations expressed entirely in the order of facts.

6.2 What joins generate

Let E, F be two rule presentations on the same catalog and let γ_E, γ_F be their closure operators with variable seeds. A set is closed under their union exactly when it is both E-closed and F-closed.

Theorem 6.2 (Alternating completion)**.** *Starting at S, set $S_{j+1} = \gamma_F\gamma_E(S_j)$. Then*

$$\gamma_{E\cup F}(S) = \bigcup_{j<\omega} S_j.$$

If the closure operators commute, the right side is $\gamma_F\gamma_E(S)$.

Proof. Each γ preserves directed unions because a finite proof tree uses finitely many seed leaves, and a common upper bound contains all their seed labels. Let $C = \bigcup_j S_j$.

For both operators, the value on S_j is contained in S_{j+1} by extensivity and monotonicity. Continuity gives $\gamma_E(C) = \gamma_F(C) = C$. Each S_j is contained in every common closed set containing S, by induction. This identifies C. If the operators commute, their composite is idempotent by their individual idempotence, giving the one-step formula. □

For $a \to b$ in E and $b \to c$ in F, the two closures of $\{a\}$ separately are $\{a, b\}$ and $\{a\}$. Their union lies below the combined closure $\{a, b, c\}$. More alternating stages are needed for the chain with rules $a_i \to a_{i+1}$ that alternate between the two presentations. The first application of $\gamma_F\gamma_E$ reaches a_2 from a_0, and each further application advances by two, until the last vertex. This follows by inspecting the only outgoing rule at each newly reached fact.

6.3 Prerequisites and regenerative proofs

Mutual implication defines an equivalence on individual facts via $a \sim b$ when $b \in \gamma(\{a\})$ and $a \in \gamma(\{b\})$. Its transitivity is cut. For unary rules with no seed source, the equivalence classes are the strongly connected components of the directed graph of rules, because unary trees are paths. For higher arity, two facts may require a common enabling environment before becoming equivalent. A base seed S gives the relative preorder $a \preceq_S b$ when $a \in \gamma(S \cup \{b\})$. Its transitivity again uses cut. Enlargement of S preserves the preorder.

A regenerative proof distinguishes a role label m from the completion label $\widehat{m}$ that records the production of a usable replacement. The initial stock contains m. A certificate for $\widehat{m}$ has a producing operation at its root. In the operational semantics, that operation deposits stock and removes the corresponding outstanding obligation. Merely reusing the seed label never discharges the obligation, because the target of the certificate is a different fact.

Proposition 6.3 (Independent modules)**.** *For a disjoint union $K = K_1 \amalg K_2$ of sets, suppose every rule takes all inputs and its output in one summand. Then $\gamma(S) \cap K_i = \gamma_i(S \cap K_i)$ for $i = 1, 2$.*

Proof. A tree whose root label is in K_i has its immediate premises in K_i. Induction on the height puts every node and seed leaf in K_i, yielding inclusion in the restricted closure. Conversely each tree in a restricted presentation is a tree in the union presentation. □

6.4 Dual certificates in finite catalogs

If r is missing from a generated set, a finite certificate of that fact is a set C containing all seeds and closed under all rules, with $r \in K \setminus C$. Soundness is containment of the least fixed point in C. Completeness follows by choosing $C = \gamma(S)$. Equivalently the missing set $M = K \setminus C$ contains r, is disjoint from the seeds, and satisfies

$$\mathrm{out}(e) \in M \quad \Longrightarrow \quad \mathrm{in}(e) \cap M \neq \varnothing$$

for every rule e. This condition says that each attempted production of a marked missing output has a marked missing input. A nullary rule makes its output unavailable as a member of M because its input intersection is empty.

A certificate of missing output can be small even when successful derivations for other outputs are large. In a chain cut at a single mandatory edge, the suffix after the edge is an admissible missing set. For branching rules, a missing set records a blocking input for every possible producing rule. When a supplement is added, the same certificate remains valid exactly when all new rules still satisfy the displayed condition and all new seeds lie outside M. This gives a constructive audit of whether a supplement is relevant to a particular shortage.

CHAPTER 7

Open processes and operadic witnesses

7.1 The operad of interfaces

We use a fixed set C of interface colors. A profile is an ordered finite sequence of inputs and a single output, denoted $(c_1, \ldots, c_n; c)$. A colored operad assigns operations to each profile, units to colors, and substitution to rooted trees of matching profiles, with symmetric actions on inputs. In spaces the operation entries are spaces and the substitutions are coherently associative and unital. Internally these entries are objects over the context B. We work with fixed colors in a slice, using a Segal presentation by colored trees. The Segal condition identifies a decorated tree with the iterated product of its vertex operation objects. This is the fixed-color version of the operadic description of composition. [5, Ch. 2][7, 8]

A symmetric monoidal category of open processes yields operations with these profiles. In an ordinary symmetric monoidal category the entry at $(c_1, \ldots, c_n; c)$ is $\mathrm{Hom}(c_1 \otimes \cdots \otimes c_n, c)$. For a symmetric monoidal higher category use its mapping space. A substitution uses the tensor of the inner arrows and then composition with the root. The coherence of the monoidal structure supplies the coherences for this construction. In a structured cospan category, gluing is implemented by pushout of interfaces. This produces categories for open networks whose boundary exchange data are available for composition. [9, 10]

Example 7.1 (Transport and assembly)**.** Let C contain f for feedstock, t for transport, p for precursor and m for assembled membrane. A transport operation $a : (t, f) \to p$ and an assembly operation $b : (t, p) \to m$ give

$$b(\mathrm{id}_t, a) : (t, t, f) \longrightarrow m.$$

The external inputs include two copies of the color t as ordered occurrences in the tree. A separate diagonal or a reusable-catalyst convention supplies a single instance to both positions. The typed operation itself specifies the two positions.

7.2 Free term presentations

A collection consists of a set or space of generators at each profile with the appropriate symmetric action. The free operad on a collection is assembled from finite rooted trees with generators at their vertices and typed, labeled leaves. Units correspond to identity edges. Composition grafts trees, and symmetric actions relabel the input positions.

Proposition 7.2 (Tree evaluation)**.** *For a collection A on C and an operad O on C, a collection map $A \to O$ specifies an operad map from the free operad $F(A) \to O$. In sets the extension is unique. In spaces the extension has a contractible space of choices over the given map.*

Proof. In sets, evaluate a generator at a corolla and evaluate an arbitrary tree by the substitutions it presents. Associativity removes the dependence on an order of successive contractions, and units identify identity edges. Thus the evaluation defines the extension. Any extension preserves grafting and has that value on each tree by induction. In the space-valued tree presentation, the Segal equivalences express the choices of a decoration at the tree in terms of the vertex decorations. Fibers of these equivalences over fixed labels are contractible. The Segal structure maps give evaluation at the contracted corolla and its coherent compatibility with grafting. This is the free-operad adjunction, with the same assignment on the generators. □

7.3 Realization images

Let $F(A) \to O$ be a physical interpretation. In a fixed profile, the image of this map is the subobject of physically interpreted operations attained by locally labeled trees. The images as a family are closed under substitution. To see this, a compatible tuple of operations in the images lifts locally to generators' trees. Because there are finitely many entries, the product of their covering families is an effective epimorphism. Graft locally and evaluate, then take the image of the cover. The identities already come from the free operad. The argument also proves the following precise minimality result.

Theorem 7.3 (Operadic generation by image)**.** *The profilewise images of $F(A) \to O$ give the smallest suboperad specified by profilewise monomorphisms that contains the given collection map. This construction commutes with geometric inverse image.*

Proof. Substitution, units, and symmetries restrict to the images by local lifting as above. The space of restrictions is contractible whenever it is inhabited, since each inclusion is a monomorphism. Thus the operadic relations and higher coherences restrict along the inclusions. A suboperad containing the generators receives the interpretation of every free tree by Proposition 7.2. It contains the profilewise images by their defining minimality. Conversely those images contain the generators, proving the universal description. Finite products of decorations, the small coproduct over tree shapes, and the colimits implementing symmetric relabeling commute with inverse image. So do the resulting images. □

For a space of possible processes with domain and codomain interfaces, passing to the image keeps the subobject of realizable processes. A further map from each such profile to a fact catalog can collapse processes with distinct operational effects at another resolution. A proof of availability for a target fact consists of a decorated tree whose

evaluation lies over that fact. The physical semantics is kept by retaining the tree and its evaluation together.

7.4 Process generators and availability guards

Each generator $g : (c_1, \dots, c_n; c)$ gives a rule on facts when its inputs and output are interpreted as availability types. There can also be an explicit guard. A guard is a subobject of the space of tuples of a process and the proposed input witnesses. Its image under evaluation is the guarded rule output. Guards that depend on the finite chosen tuples can be included by restricting the family of admissible decorated trees at each construction stage. When guards refer to quantitative state, we instead retain the tuple as part of a transition, as in Chapter 16.

Corollary 7.4 (A guarded construction)**.** *For a presentation with a fixed seed family, a fixed rule family, and a fixed subfamily of admitted rules in each arity, the guarded production image is given by Theorem 5.3 with the admitted rule families. If the admission relation enlarges, so does the generated subobject.*

Proof. Apply the theorem to the admitted rule families with their inherited input and output maps. The inclusion of admitted families induces the inclusion of tree images by the comparison of presentations. □

7.5 Two projections of a proof

A tree with labels in a rule catalog has a root capability and a list of free interfaces. When the leaves are all supplied by designated seeds, the root capability lies in the generated subobject. When some leaves are free, the tree is a conditional operation on those interfaces. Composition substitutes witnesses for free interfaces. In sets, a free tree gives the conditional rule $L \rightsquigarrow c$, where L is its set of free leaf labels and c is the root. The exact input profile retains occurrences and their symmetric ordering in addition to the labels in L.

If a conditional tree is evaluated using another family of trees at its free leaves, the result is a proof of its root. This is the tree-level proof of cut from Chapter 6. Cost, source labels, and inventory effects can be read from the substituted trees before projecting to facts. Thus one graphical operation carries the logical rule, the compositional process, and the quantitative record.

CHAPTER 8

Coupling along interfaces

8.1 The universal comparison

Let $P \leftarrow I \rightarrow H$ be maps of operads on C. Define the coupled operad

$$P \star_I H = P \amalg_I H.$$

In spaces use the homotopy pushout. An interface operation has two interpretations, and the coupling supplies their coherent agreement. For any operad O, the defining property is the equivalence

$$\mathrm{Map}_{\mathrm{Op}}(P \star_I H, O) \simeq \mathrm{Map}_{\mathrm{Op}}(P, O) \times_{\mathrm{Map}_{\mathrm{Op}}(I,O)} \mathrm{Map}_{\mathrm{Op}}(H, O).$$

The right-hand homotopy fiber product remembers a path of interpretations of the interface. A physical realization is such a compatible pair into a physical process operad. The induced map out of the pushout can identify formal composites that have the same effect there.

8.2 Mixed terms in the ordinary pushout

Theorem 8.1 (Presentation of the coupled operad)**.** *For set-valued fixed-color operads the coupled operad is generated by labeled trees with vertices in P or H. Its relations are the symmetric operad relations, evaluation of a subtree wholly in one side, and identification of the two images of operations in I. Under any compatible interpretation into O, the image consists exactly of the evaluations of such mixed trees.*

Proof. Form the free operad on the coproduct of the underlying collections. Quotient by the congruence generated by the listed relations, including the unit relations of each operad. The corolla maps from P and H now respect their internal substitution, units, and symmetries. They agree on I. Any compatible pair into O extends uniquely from generators by Proposition 7.2. Evaluation respects the relations, so the extension factors uniquely through the quotient. This gives the pushout universal property. The map from the free operad to its quotient is surjective profilewise, so the evaluated image has exactly the stated representatives. □

Example 8.2 (A strict mixed output)**.** Let the colors be f, m, r. Take P to be free on $v : m \rightarrow r$, H free on $h : f \rightarrow m$, and I the identity operad. The maps out of the pushout are arbitrary choices of interpretations of v and h, so the pushout is free on the pair. Its profile $(f; r)$ contains $v(h)$. The corresponding entries in each side are empty. Interpreting v as use of host machinery and h as production of that machinery exhibits a strictly new physical role when the interpretation of $v(h)$ is retained.

8.3 Coherent realizations

For an operad in spaces, let $\pi_0 P$ denote the operad obtained by taking components at every profile. The construction respects products of spaces and sends the unit and substitution maps to the corresponding maps of components. The Segal equivalences therefore give the operad laws on components. It is left adjoint to the discrete inclusion. Indeed, in a map to a discrete operad the map at each profile factors uniquely through the components, and the operation laws are checked on those components. This proof works internally with 0-truncation into the slice's ordinary topos of 0-truncated objects. Finite products of profile objects are preserved by that truncation in the slice. For a 0-truncated Z, its exponential Z^Y is 0-truncated because maps into it are maps into Z after taking a product with Y. The adjunction for truncation can therefore replace either factor of a product before mapping to Z, which proves preservation of finite products. Equivalently, the component construction is performed by applying 0-truncation levelwise in the fixed-color Segal presentation.

Corollary 8.3 (Components of coupling)**.** *Taking components of a fixed-color operadic homotopy pushout gives the pushout of the component operads. In particular over a point the components of the coupled operation spaces are represented by mixed terms modulo the relations of Theorem 8.1.*

Proof. The components functor preserves colimits by the adjunction. Apply Theorem 8.1 to the resulting pushout in sets. □

The homotopy pushout itself records higher information about the relation between those terms. Its mapping property records the choices of interface agreement as a space. Consider identity-colored unary operads with a single color. They are associative monoid objects in spaces. Maps from an interface free on one unary generator are parameterized by the target's unary operation space. A compatibility path is therefore a path between the two candidate images of that generator. Where those images coincide at a point o, the space of such paths is the based loop space at o. This gives a simple instance where specifying equal component classes leaves a nontrivial space of couplings.

8.4 The two images of a coupling

For a coupled pair P, H, there are two computable images at a target catalog. The first comes from operations in either side individually. It is their join by Lemma 3.1. The second comes from the entire pushout. The first maps into the second because the two maps to the physical operad factor through the pushout. Example 8.2 has a strict inclusion at profile $(f; r)$. A coupled supply can therefore have a strict gain over the join of separate supply images, even while individual cost bounds and type constraints are respected at every vertex.

Theorem 8.4 (Descent of coupled supply)**.** *Let a fixed-color coupling and a physical interpretation be given over B in the internal tree presentation. For $T \to B$, their base changes give the same profilewise image as pulling back the coupled realization image. For an effective epimorphism $T \to B$, closure of a required profile family holds over B exactly when it holds over T.*

Proof. Slice pullback is strong cartesian monoidal and preserves limits and colimits. Its right adjoint preserves finite products. The adjunction induces the corresponding adjunction for fixed-color operadic diagrams and their pullbacks. In the Segal presentation, preservation of products gives the Segal condition after pullback. Thus pullback sends the operadic pushout to the pushout of the restricted diagram. It preserves the effective epimorphism and monomorphism factors of each realization map. The equivalence of the closure judgments follows from Theorem 3.4 on each profile. □

8.5 Change of boundary by nested coupling

Suppose a first enlargement adjoins H through I, and a second adjoins J through an interface L whose map lands in $P \star_I H$. The universal property of the final operad is a tuple of three interpretations with agreement along I and along L, with the second agreement computed from the first interpreted pushout. Iterating the mapping-space formula proves this assertion. In particular, when $L \to P \star_I H$ factors through H, one obtains a canonical equivalence

$$(P \star_I H) \star_L J \simeq P \star_I (H \star_L J).$$

The two iterated mapping-space limits impose exactly the same agreement conditions, and Yoneda gives the equivalence. This is the associativity of enlarging a boundary along compatible interfaces. A supplied manufacturing stage can be grouped with a factory or with the device under maintenance by this comparison.

CHAPTER 9

Stocks, catalysts, and regeneration

9.1 From availability to multiplicity

For a finite species set C, a stock is a vector $x \in \mathbb{N}^C$. A transition has an input $u \in \mathbb{N}^C$ and an output $v \in \mathbb{N}^C$. It is enabled at x exactly when $x \geq u$ coordinatewise and has successor $x - u + v$. A catalytic use of species c places its units on both sides. Enabling still tests the stock of c, while the net change in that coordinate is zero. A consumptive use can instead reduce that coordinate.

The underlying availability rule takes the support of u to the support of v, using one conclusion at a time. This projection is sound for reachable stocks relative to the union of facts derived from the initial support. Each enabled transition has its prerequisite facts in the stock's support, hence in the derived set by induction. Newly positive coordinates are among the outputs. This gives the claim at the successor stock.

Example 9.1 (A multiplicity threshold)**.** Start with one m, and take a single reaction $2m \to y$. The Boolean rule derives y from m. The stock graph at the initial point has no enabled production edge for y, since the coordinate of m is 1. With two units of m the edge is enabled and deposits y. The stock graph records the threshold at 2, while the Boolean rule records the role of m.

9.2 Exact sequence certificates

For an ordered list $e_1, \ldots, e_n$ of reactions, write u_i, v_i for its inputs and outputs, and $d_i = v_i - u_i \in \mathbb{Z}^C$. Set

$$h_j = \sum_{i<j} d_i, \qquad q = \bigvee_{1 \leq j \leq n} (u_j - h_j) \ \vee \ 0,$$

where the joins are coordinatewise maxima. For the empty sequence use $q = 0$.

Theorem 9.2 (Minimal enabling stock)**.** *The sequence is executable from x exactly when $x \geq q$. In that case its final stock is $x + \sum_i d_i$.*

Proof. Before step j an executed prefix has stock $x + h_j$. Its next enabling condition is $x + h_j \geq u_j$, or $x \geq u_j - h_j$. These inequalities for all j, together with $x \geq 0$, are exactly $x \geq q$. Conversely this condition ensures every step is enabled by induction on the prefix. The final stock is the cumulative update. □

The pair (q, d) consisting of the minimal stock and net change of a sequence determines its partial action on the stock space. For a first sequence with pair (q_1, d_1) followed by a

second with (q_2, d_2), the combined pair is

$$\Big(q_1 \vee (q_2 - d_1),\ d_1 + d_2\Big).$$

The two enabling inequalities are $x \geq q_1$ and $x + d_1 \geq q_2$. They give the formula. The identity has pair $(0, 0)$. Every sequence pair satisfies $q + d \geq 0$. Associativity is both a direct consequence of distributivity of translation over maxima and a consequence of associativity of composition of partial maps.

9.3 Repeatable stocks

Corollary 9.3 (Repeatability of a prescribed sequence)**.** *For a sequence with pair (q, d) and finite initial stock, indefinite repetition is possible exactly when $x \geq q$ and $d \geq 0$ coordinatewise. With a coordinatewise capacity ceiling on all states, an indefinitely repeated, unsaturated sequence has $d = 0$.*

Proof. The sequence starting at the k-th round sees the stock $x + kd$. If $d \geq 0$ and $x \geq q$, all these stocks dominate q. If a coordinate of d is negative, that coordinate falls below q for large k. With an upper bound, a positive coordinate of d eventually exceeds it, so all coordinates of d must vanish. □

A repeated repair sequence with zero net stock change still executes the work of restoring each component. For example, $m + f \to p$ followed by $p \to m$ has one unit of net feedstock consumption. A round that imports a feedstock first gives the zero vector over the full stock, and its minimal initial stock contains the preexisting m. The label on the final reaction records repair. The proof of repeatability then checks both replacement and regeneration at the same boundary.

9.4 A resource interface as a partial action

A boundary interface can be described by the sequence certificates it admits. The formula for (q, d) presents a monoid of stock transformations, with order on their domains given by q. Transformations with common net change d can be compared by their minimal stock thresholds. The one with smaller q is defined on a larger set of stocks. Composition preserves this order in each argument because translation and coordinatewise maximum are monotone.

When a host operation reduces a threshold without changing the ultimate stock effect, the order comparison is exact on the entire stock domain. When it supplies a new effect, the two transformations are compared by their output and domain pairs. This provides a quantitative refinement of the image comparison of Chapter 3, using data that can be evaluated before running the state dynamics.

CHAPTER 10

Budget grades and attainable realizations

10.1 Cost of a realization

A scalar charge is an element of $[0, \infty]$ with addition and zero. A vector of charges is in $[0, \infty]^d$, with componentwise addition and order. An ordered commutative monoid of charges M has an order monotone under addition. We use its displayed order for upper bounds. When referring to Lawvere enrichment we reverse this order, so that a cheaper realization has a larger enriched value. [13]

A ledgered operad is an operad of realizations O_{r} with a charge map θ on each profile. The charge of a unit is zero. A composite has charge at most the sum of the charges of its factors. A strictly additive ledger has equality. Let $p : O_{\mathrm{r}} \to O$ send a realization to its abstract operation, preserving compositions. The set

$$A(f) = \{\, b \in M \mid \exists\, \widetilde{f}, \quad p(\widetilde{f}) = f, \quad \theta(\widetilde{f}) \le b \,\}$$

is an upper set of attainable budgets for f. With operation objects in a topos, define the attainable relation by taking the image of the object of pairs $(\widetilde{f}, b)$ satisfying the budget bound. The same expression is then read internally. This makes attainment a positive existential relation over the profile and the budget.

Theorem 10.1 (Composition of budgets)**.** *If $b \in A(f)$ and $b_i \in A(g_i)$ for every i in a compatible input profile, then $b + \sum_i b_i \in A(f(g_1, \dots, g_n))$. A tree with vertex budgets b_v is realizable at their sum.*

Proof. Choose the indicated realizers, substitute them, and use the charge inequality and monotonicity of addition. In the internal reading, a finite family of witnesses is chosen over a common effective cover, obtained by the product of their covers. The composite is a witness over that cover, so its budget lies in the image. Induction on the root and branches gives the formula for a tree. The empty tree uses a unit and has charge zero. □

10.2 The graded operad

For each budget b, the realizations of charge at most b form a collection. With varying budgets these collections have maps induced by monotone enlargement and compositions that add budgets. More explicitly, regard a budget as an extra index of a lifted operation, and send

$$(\widetilde{f}, b), \quad (\widetilde{g}_i, b_i) \;\longmapsto\; \Big(\widetilde{f}(\widetilde{g}_i),\, b + \textstyle\sum_i b_i\Big).$$

Associativity uses associativity of substitution and of addition. The unit is the identity with budget 0. Symmetric actions permute summands and preserve the bound. This gives an M-graded operadic structure. A budget-b collection can compose with another budget-b collection at the larger grade given by the addition rule.

For example, take a free unary operad on f with charge 1 and the additive ledger. Grade 1 includes f, and the resulting composite f^2 appears at grade 2. More generally the operations f^k have least grade k, by induction on the number of letters. Thus successive substitutions require the budget index to change. A single fixed budget defines feasibility queries about a collection of operations.

10.3 Infimum values and exact feasibility

For scalar charges put $c(f) = \inf\{\theta(\widetilde{f}) \mid p(\widetilde{f}) = f\}$, with empty infimum ∞. For composable operations the inequality

$$c(f(g_1, \dots, g_n)) \le c(f) + \sum_i c(g_i)$$

holds. When all summands on the right are finite, pick realizers within $\varepsilon/(n+1)$ of the indicated values and apply Theorem 10.1. Sending ε to zero gives the inequality. When a summand is infinite the right side is top in the numerical order. In the unary case the resulting costs form a Lawvere-enriched comparison on objects after taking the infimum also over abstract arrows between them. Identity costs zero and the triangle inequality follows by optimizing over composable arrows.

Example 10.2 (An open attained-budget ray)**.** Let a unary operation have realizations of charges $1 + 1/n$ for $n \ge 1$. The attainable finite budgets are $(1, \infty)$, and $c(f) = 1$. The budget 1 is outside $A(f)$. Each $b > 1$ contains a witness by choosing n with $1/n \le b - 1$. A second operation with a realization of charge 1 has the same infimum and contains 1 in its budget set.

For vector grades a coordinatewise infimum can lose even more information. The realizations $(1, 4)$ and $(4, 1)$ have coordinatewise infimum $(1, 1)$, and their attainable budgets are the union of the two orthants above the given points. The vector $(2, 2)$ belongs to neither orthant. Keeping the upper set preserves exactly the feasibility information for all queries.

10.4 Budgeted trees in context

Let the rule presentation of Chapter 5 carry a grade for each seed and each rule. Grade the tree by summing its leaf and vertex grades. At height j, let $T_j^{\le b}$ be the family of trees with grade at most b. Its image $G_j^{\le b}$ gives availability at that height and budget. The eventual image is

$$G^{\le b} = \bigvee_j G_j^{\le b}.$$

The restriction of the tree family to budgets is a pullback, and the same local-match argument gives closure for $G^{\leq b}$. At a variable budget b the entire construction is expressed by an attainable relation in $K \times_B M$. For any geometric inverse image transporting the monoid, its order relation, and the grade maps, this attainable relation is transported as well. The proof is preservation of the products, images and colimits used to build the relation. The exact query $r \in G^{\leq b}$ therefore admits the same descent argument as ungraded availability.

Proposition 10.3 (Nested budget regions)**.** *If $b \leq b'$, then $G^{\leq b} \leq G^{\leq b'}$. If every primitive grade is zero and $0 \leq b$, then $G^{\leq b} = G$.*

Proof. Every witness satisfying the smaller bound satisfies the larger bound. A zero-grade sum is zero, so under the second condition every derivation is admitted at the given budget. □

10.5 A stock budget and a path budget

The sum of primitive exergy charges measures a completed tree under an additive boundary ledger. The ability to execute that tree is a question about when exergy is available along a schedule. A nonnegative total charge gives a static lower bound. A timed ledger also checks the balance before each operation. Consider two operations of charges 2 and 5, in that order, with arrivals 0 and 7. The total supply equals the total charge. An initially empty battery fails the first guard, while an initially charged battery with two units executes both steps, subject to a capacity of at least five units under arrival-first accounting. The static inequality and the timed feasibility test refer to the same total but use different state information.

CHAPTER 11

Optimal finite proofs and portfolio costs

11.1 Height-bounded optimization

Let K be a finite set, with finitely many rules and seeds carrying finite nonnegative scalar charges. Denote by $c_j(k)$ the least charge of a proof of height at most j, with ∞ for a missing proof. At stage zero, $c_0(k)$ is the smallest charge of a seed labeled k, with the same infinity convention. For $j \geq 0$,

$$c_{j+1}(k) = \min\left(c_j(k),\ \min_{e:\ \mathrm{out}(e)=k}\left[w(e) + \sum_{a \in \mathrm{in}(e)} c_j(a)\right]\right).$$

The sum uses the ordered premise list with its multiplicities. It is zero for a nullary rule. This is a finite dynamic program over the height of the tree. It applies when the same subproof can be copied and its charge is charged for every occurrence.

Theorem 11.1 (Scalar optimality and attainment)**.** *The recurrence gives the least charge at every height. For $N = |K|$, its value at $j = N$ is the least charge over all finite proofs. Every finite value at that stage is attained by a tree.*

Proof. A tree of height at most $j + 1$ is an already admitted tree, a seed, or a rule with each branch of height at most j. With a fixed root rule the choices of branches are independent, so the minimum is the sum of their minima and the root charge. All stagewise candidates are finite in number. Induction proves the recurrence and stagewise attainment.

If a root-to-leaf path repeats a fact label, replace the subtree at its higher occurrence by the subtree at its lower occurrence. It has the same output label, so every parent remains well typed. The replacement deletes nonnegative charges, including those of side branches. It strictly decreases the number of vertices. Repeating the operation yields a tree without a repeated fact along a branch and of height at most N, with charge at most that of the original tree. Thus every proof has a competitor at height at most N. The finite stage contains a minimum among these competitors and its members are genuine proofs. □

The pruning step compares proofs with the same full fact label. Including an interface state, timing class, or obligation in that label narrows the possible replacements. When all such data are included in a finite label set, the theorem applies with N equal to the size of that enlarged set. An energy ledger represented outside the label set is checked separately in the operational compilation.

11.2 Pareto recurrence

For a finite family of proof trees with grades in $\mathbb{R}^d_{\geq 0}$, keep only the componentwise minimal grades. When two grade vectors coincide, one stored certificate suffices for feasibility, and the realizations with that grade may still be retained for other queries. Write $\min_{\leq}$ for this Pareto reduction. In the recurrence, take unions over candidates and Minkowski sums along premises, then apply $\min_{\leq}$.

Proposition 11.2 (Exact Pareto computation)**.** *The Pareto recurrence computes all minimal proof grades. For a finite presentation on N labels and nonnegative grades, height N suffices. A budget admits a proof exactly when it dominates at least one retained grade.*

Proof. Replacing a grade in a sum by a smaller grade makes the sum smaller. Thus removing dominated grades at one stage leaves the upward closure of all candidate sums at the next stage unchanged. Induction shows that each stage represents the upward closure of all its proof grades, with every retained grade realized. The repeated-label pruning argument from Theorem 11.1 lowers every coordinate, giving a dominating approximation from below at height N for any proof grade. Equality of the upward closures proves the result and the budget test. □

Take a fact m with two seed realizations of grades $(2, 5)$ and $(5, 1)$, and a fact t with a seed of grade $(1, 2)$. A rule from (m, t) to a repair costs $(1, 0)$. The repair grades are $(4, 7)$ and $(7, 3)$. A direct repair of grade $(6, 8)$ is dominated by $(4, 7)$. The minimal list is exactly the two incomparable grades. A balance $(6, 5)$ admits neither proof, although it exceeds the coordinatewise infimum $(4, 3)$. A balance $(7, 5)$ selects the second grade.

11.3 Simultaneous demand

Independent witness branches can be combined with a new target z and a zero-charge rule whose premise list is the desired list of targets. A grade for z is then the sum of their selected proof grades. This compilation implements simultaneous provisioning when the charge model counts repeated use. If an operation supplies two roles from a shared intermediate, a joint operation for the pair can be entered as a primitive or a fact encoding that paired output. The optimizations are performed over the explicitly represented sharing. Reusing a Boolean assertion of availability never changes the additive grade of a separately chosen branch.

Theorem 11.3 (Optimal sources with a budget)**.** *In a finite presentation with a finite source set J, attach to each tree its pair (I, w), where $I \subseteq J$ is its source set and $w \in \mathbb{R}^d_{\geq 0}$ its grade. With the product order of inclusion and componentwise comparison, the minimal pairs are computed by the height-N recurrence using union on sources and addition on grades. A portfolio S and a budget b supply a target exactly when a retained pair lies below (S, b).*

Proof. The product order is preserved by union and grade addition. This proves the stagewise reduction using the argument of Proposition 11.2. Pruning a repeated-label branch removes source labels and lowers grades, so height N suffices. A retained pair comes from a tree. Such a tree is available with S, b exactly when $I \subseteq S$ and $w \leq b$. Conversely a tree satisfying those conditions dominates a retained pair by finiteness and the stage bound. □

The conclusion separates the acquisition cost of a supplement from the running cost of its operations. A one-time cost assigned to a chosen source portfolio is evaluated on S, and the operation grade is tested against b. The pair calculation allows these costs to be evaluated independently until a design criterion couples them.

CHAPTER 12

Exergy ledgers and finite resources

12.1 A boundary balance

A discrete ledger records stored usable exergy e_j, supply s_j, operation charge c_j, and exported or spilled exergy o_j over the j-th interval. Each is nonnegative. Conservation of the modeled currency gives

$$e_{j+1} = e_j + s_j - c_j - o_j.$$

The operational rule below will separate arrival, clipping, and use. For cumulative estimates the balance alone suffices. The physical interpretation is that a boundary fixes which transfers are supplies and which transfers cancel internally. Work drawn by a component of a larger coupled system appears in that component's ledger as an input and at the delivering component as an output. Summing balances over the two components cancels the transfer.

Theorem 12.1 (Cumulative maintenance bound)**.** *For n intervals of a ledger, one has*

$$\sum_{j<n} c_j \le e_0 + \sum_{j<n} s_j.$$

If every admissible maintenance interval costs at least $\eta > 0$, a total finite available supply $E = e_0 + \sum_{j\ge 0} s_j$ permits at most $\lfloor E/\eta \rfloor$ completed intervals.

Proof. Sum the balance and solve for the cumulative charge. The terminal stock and exported amounts are nonnegative, giving the inequality. The charge of n completed intervals is at least $n\eta$, so $n \le E/\eta$, and the integer bound follows. □

A designated repair performed once per interval supplies a uniform bound. Let its least realized charge be η, with all other charges nonnegative. Then every execution of the interval has charge at least η. This uses a lower bound on required events, computed in the production model. The argument applies to the set of finite successful intervals directly.

12.2 Throughput with a bounded initial store

Proposition 12.2 (Long-run throughput test)**.** *Suppose for every n a successful history of n intervals has cumulative charge at least $nm - C$, where $m \ge 0$ and $C \ge 0$ are fixed. If the supply sequence satisfies $\limsup_{n\to\infty} n^{-1} \sum_{j<n} s_j < m$, the lengths of successful histories from a fixed initial store are bounded.*

Proof. Choose $\delta > 0$ with the limsup below $m - \delta$. For all sufficiently large n, the cumulative supply is at most $n(m-\delta)$. Theorem 12.1 then yields $nm - C \le e_0 + n(m-\delta)$, or $n\delta \le e_0 + C$. This bounds all sufficiently large successful lengths, and the finite remaining lengths also have a bound. □

The estimate uses a burden defined on finite maintenance histories. A replenishing loop with charge c and per-step supply s shows the two outcomes directly: $s < c$ forces the number of repetitions to be finite for a finite initial store, while $s \ge c$ and sufficient capacity admit the loop repeatedly under the arrival-first rule. If the rule's order charges the step before receiving its supply, starting stock at least c supplies the first guard and the end stock is replenished. The moment of arrival is part of the dynamics.

12.3 Combining ledgers

Suppose systems P, H exchange a quantity t_j during interval j, credited to P and debited from H. Add the two balances. The combined system has initial store $e_0^P + e_0^H$, external supplies $s_j^P + s_j^H$, external outputs $o_j^P + o_j^H$, and operation charges $c_j^P + c_j^H$. The term t_j disappears. The same cancellation applies to several interfaces by summing the two incidences of each internal transfer. When a transfer is lossy, credit only $t_j - \ell_j$ to the recipient for loss $\ell_j \ge 0$. The total balance contains the additional charge ℓ_j.

Example 12.3 (Host-supplied work)**.** A dependent subsystem must spend three units at each repair. It receives two units per interval from a host and one unit through its environmental interface. The host expends one unit on its own maintenance and transfers two. The subsystem ledger records a throughput of three, the host ledger requires a throughput of at least three, and the combined ledger requires a throughput of four. These numbers follow by applying the balance to each subsystem and canceling the internal two-unit transfer.

12.4 Batches and recurring burdens

Let c_i be a lower bound for the charge of each occurrence of obligation i. If a legal history of length n must realize at least $\ell_i(n)$ separately charged occurrences, additivity gives $\sum_{j<n} c_j \ge \sum_i \ell_i(n) c_i$. For prescribed positive asymptotic frequencies λ_i with $\ell_i(n) \ge n\lambda_i - a_i$, the throughput theorem applies with $m = \sum_i \lambda_i c_i$ and $C = \sum_i a_i c_i$. Joint repair operations that discharge more than one obligation can instead be represented as one charge with a vector of discharged counts. A valid linear lower bound on the vector of counts produces a burden by summing over those joint operations. Chapter 24 gives a dual certificate of this kind.

CHAPTER 13

Dissipation in explicit process models

13.1 Flux and affinity

We use entropy production and exergy in distinct roles. Exergy is available work relative to an environment. With environmental temperature $T_0 > 0$, the destroyed exergy associated with entropy production Σ is $T_0\Sigma$ in a balance using that environment. A transition charge can include dissipation and other nonnegative allocations required by its resource model. Open reaction networks supply explicit examples in which driving, storage, and dissipation are separately accounted. Stochastic thermodynamics extends these balances to trajectories of mesoscopic systems. [28, 29]

For a reversible reaction channel with positive forward and reverse fluxes a, b, put $J = a - b$ and use the dimensionless affinity $\log(a/b)$. The channel's dimensionless production is

$$J \log(a/b) \geq 0.$$

Since the logarithm is increasing, the two factors have the same sign. The value is zero exactly when $a = b$. Multiplying by Boltzmann's constant for particle-scale fluxes, or using consistent molar units with the gas constant, gives entropy per unit time. Summation over channels gives a nonnegative rate. A channel with external chemical forcing can have a persistent nonzero current even when the stock of an internal species is stationary.

13.2 Finite-state Markov balance

Consider a continuous-time chain with rates $k_{ij} > 0$ on the edges of a finite undirected support and positive state probabilities $p_i(t)$. Write $a_{ij} = p_i k_{ij}$ and $J_{ij} = a_{ij} - a_{ji}$, and let $i < j$ select an orientation of each supported edge. The master equation is $\dot{p}_i = -\sum_j J_{ij}$. For Shannon entropy in units where Boltzmann's constant is one,

$$\frac{d}{dt}\left(-\sum_i p_i \log p_i\right) = \sum_{i<j} J_{ij} \log(p_i/p_j).$$

Differentiate, use $\sum_i \dot{p}_i = 0$, and collect the two oriented edge contributions. Define the entropy-flow contribution $\dot{S}_{\mathrm{e}} = \sum_{i<j} J_{ij} \log(k_{ij}/k_{ji})$ for the reservoirs in a local detailed-balance interpretation. Then

$$\dot{S}_{\mathrm{tot}} = \sum_{i<j} J_{ij} \log(a_{ij}/a_{ji}) = \frac{d}{dt}\left(-\sum_i p_i \log p_i\right) + \dot{S}_{\mathrm{e}}.$$

The channel inequality proves nonnegativity and characterizes zero total production by equality of the two fluxes on every edge. This is detailed balance at the stated distribution. Additivity holds for independent networks with summed entropy ledgers. [29, 11]

Proposition 13.1 (A driven stationary cycle)**.** *On the cycle of length $m \geq 3$, let the clockwise rate be $u > 0$ and the counterclockwise rate be $v > 0$ at every vertex. At the uniform distribution the total production is $(u - v)\log(u/v)$. The distribution is stationary. The production is positive exactly when $u \neq v$.*

Proof. At every vertex the incoming total and outgoing total are both $(u + v)/m$, so the master equation vanishes. Along a clockwise edge the fluxes are u/m and v/m, hence the production is $(u - v)\log(u/v)/m$. Summing the m equal contributions gives the formula. Strictness is the strict monotonicity of the logarithm. □

With $u = 3$ and $v = 1$, the production is $2\log 3$. The per-edge current on a three-state cycle is $2/3$. With $u = v = 2$, the same distribution is stationary and the production is zero. Thus the state constraint that the distribution be uniform admits two process realizations with different budgets. A viability predicate on states and a ledger on transitions record both pieces of data.

13.3 Free energy for a detailed-balance chain

Let π be a strictly positive detailed-balance distribution for the fixed rates, so $\pi_i k_{ij} = \pi_j k_{ji}$. Define $F(p) = \sum_i p_i \log(p_i/\pi_i)$. Then

$$\dot{F}(p) = -\sum_{i<j} J_{ij} \log\left(\frac{p_i/\pi_i}{p_j/\pi_j}\right) = -\dot{S}_{\mathrm{tot}}.$$

For the first equality differentiate and collect edge terms. Detailed balance makes the logarithm the flux ratio logarithm, proving the second. Thus F is nonincreasing and decreases strictly at positive production. If the support graph is connected, equality at a time implies the ratios p_i/π_i are constant along every edge, hence $p = \pi$ by normalization. If the graph has several components, equality fixes the ratio on each component, with its component mass conserved by the master equation.

13.4 A charge from a required current

For the fixed biased cycle of Proposition 13.1, maintaining the uniform stationary distribution with that generator for a time interval Δ produces $\Delta(u - v)\log(u/v)$ units of dimensionless entropy. At temperature T_0, choose units with Boltzmann's constant one and assign destroyed exergy $T_0\Delta(u - v)\log(u/v)$. A maintenance schedule that repeatedly invokes this specified process has that exergy term in every interval. If the only usable exergy initially available is E and the balance has no supplies, Theorem 12.1

bounds its completed intervals by the corresponding quotient. This proof takes the current from the operation's specification and the burden from the boundary balance.

The optimization of control costs for specified nonequilibrium target states studies more general constrained interventions on Markov systems. The controlled process, target, and available interventions enter those bounds. In the algebra developed here, replacing one fixed driving process by an available family corresponds to enriching its operation profile with charged realizations and optimizing over their feasible schedules. [30, 28]

13.5 Open-network accounting at composition

A chemostat maintains a reservoir parameter while allowing net material exchange. For a model with a specified external chemical-work term, an internal free-energy change, and a nonnegative dissipative term, concatenate two intervals at their common state. The changes in internal free energy telescope and the work and dissipation add. For a parallel composition with a shared reservoir, the total flux to the reservoir is the sum of its incidences. With a fixed reservoir potential, the corresponding power is linear in that total flux. When the reservoir becomes internal, its balance must be added and its stock becomes a state coordinate. This is the thermodynamic version of the coupling in Chapter 8.

A model for repair can therefore carry a two-part grade. One coordinate measures reserved electrical or chemical work and another measures dissipation along the path. Additive grades combine the charges across independent realizations. An order test on these coordinates implements a budget on both. Projection to a single scalar optimizes a prescribed linear functional of the two charges. The Pareto calculation of Chapter 11 keeps all joint tradeoffs.

CHAPTER 14

Viability as a greatest fixed point

14.1 Relations and safe predecessors

In $\mathcal{E}_{/B}$, let S be the state object, $V \hookrightarrow S$ the safe subobject, and L the object of admissible transitions, with source s and target t. Labels can include a control, a realization of a repair, a disturbance, or a full ledger. For $A \hookrightarrow S$ put

$$\mathrm{Pre}(A) = \mathrm{im}\Big(L \times_{S,t} A \xrightarrow{s} S\Big), \qquad \Psi(A) = V \wedge \mathrm{Pre}(A).$$

Thus $\Psi(A)$ contains a safe state when that state locally admits an outgoing transition into A. The direct-image formula makes Ψ monotone. Both its operator and its geometric presentation are distinct from the production operator Φ. [16, 17, 14]

Theorem 14.1 (Viability by coinduction)**.** *The subobject $K = \bigvee\{A \hookrightarrow S \mid A \le \Psi(A)\}$ satisfies $K = \Psi(K)$. It is the largest safe subobject for which the object of internal transitions with source and target in it covers it by source.*

Proof. Every postfixed A is below K, so $A \le \Psi(A) \le \Psi(K)$. The join of all such A yields $K \le \Psi(K)$. Monotonicity now gives $\Psi(K) \le \Psi^2(K)$, which puts $\Psi(K)$ among the postfixed subobjects and hence below K. For the second assertion, a subobject A is postfixed exactly when $A \le V$ and $A \le \mathrm{Pre}(A)$. Pulling the predecessor image back to A identifies the latter with the image of the internal transitions $A \times_S L \times_S A \to A$ being all of A. □

The calculation is the greatest-fixed-point part of the lattice theorem of Tarski. The coinduction rule certifies inclusion in K from a subobject that stays within V and supports a local successor in itself. A trajectory gives such a certificate when its states are collected into a subobject and its successive edges supply their successors. More generally the certificate may cover each state by different local edges. [12]

14.2 Finite horizons and transfinite continuation

Set $V_0 = V$, $V_{\alpha+1} = \Psi(V_\alpha)$, and $V_\lambda = \bigwedge_{\alpha<\lambda} V_\alpha$ at a limit ordinal. This chain is descending by monotonicity and the initial safety meet. It eventually stabilizes since $\mathrm{Sub}(S)$ is a set. Once $V_{\alpha+1} = V_\alpha$, the common value is a fixed point. It is the greatest fixed point because every postfixed point lies below every stage by transfinite induction. The finite stages have a path interpretation.

Lemma 14.2 (Finite paths)**.** *For $n \in \mathbb{N}$, the subobject V_n is the source image of the object of paths of length n with all $n+1$ states in V.*

Proof. For $n = 0$ the path object is V. At the next stage, pull the object of safe n-step paths back along the target of one edge whose source lies in V. The image under source is $\Psi(V_n)$, since pulling back the image factorization of the tail path map gives the same image as first using the tail object. This is the desired path object with one added step. □

In sets, take a state x with outgoing edges to the top of every finite descending chain, and let each chain end at a dead end. Put all states in V. The state x belongs to V_n for every finite n. Every other state has a finite remaining length, hence $V_\omega = \{x\}$. There is no transition from x to itself, so $V_{\omega+1} = \varnothing$, the greatest fixed point. The intersection of finite horizons can therefore be computed before the greatest fixed point has stabilized.

14.3 Finite branching and exact histories in sets

Proposition 14.3 (Finite branching theorem)**.** *For a set-valued graph with finite out-degree at every safe state, $K = \bigcap_{n<\omega} V_n$. With the ordinary set-theoretic choice of successors, its members are exactly the sources of infinite safe paths.*

Proof. Write $A = \bigcap_n V_n$. At $x \in A$, the sets of safe successors in V_n are a nested sequence of nonempty subsets of a finite set, because $x \in V_{n+1}$ at every stage. Their intersection is nonempty. Thus $A \leq \Psi(A)$, proving $A = K$. A chosen successor in K at each state gives an infinite path by recursion. Conversely the set of states along an infinite safe path is postfixed, hence contained in K. □

14.4 Descent of the kernel

Slice pullback along $u : T \to B$ commutes with images and with arbitrary meets of subobjects. For the latter fact, pullback between slices has both adjoints, so it preserves limits, and a meet of subobjects is their fiber product over the ambient object. Pulling the predecessor formula back gives the predecessor for the restricted graph. Consequently it transports every stage of the transfinite chain, by induction including limit ordinals. Choose an ordinal at which the chain over B stabilizes. Its transported stage stabilizes the chain over T, and the proof of greatestness applies there. Thus $u^*K = K_T$.

For an effective cover $T \twoheadrightarrow B$, local inhabitation of K over B is equivalent to local inhabitation of K_T over T. Preservation and reflection of the context image follow as in Theorem 3.4. The kernel is computed using the same transition regime on each pullback. When a coupling adds transitions, the order relation instead compares two kernels on a common state object.

Theorem 14.4 (Uniform finite descent)**.** *Let the state object over B be locally constant finite of cardinality at most N. Then $K = V_N$. Every geometric inverse image transports this kernel and its context image.*

Proof. By the preceding descent argument, the assertion about stabilization is local on

B, so trivialize the state object along an effective cover. A length-N safe path on at most N states contains a repeated vertex. It therefore has a safe extension of length $N + 1$ starting at its original source, obtained by following the prefix to the repeated vertex and traversing its nonempty loop enough times. The finite pigeonhole argument concerns decidable labels in a constant finite object and holds internally. By Lemma 14.2, this gives $V_N \leq V_{N+1}$. The chain gives the reverse. The argument applies to the pulled-back graph under any geometric inverse image as the local finite bound is preserved. The formula for V_N uses finite limits and images, so it commutes with that inverse image, as does its context image. □

14.5 A point that sees an unbounded horizon

For sheaves on $\mathbb{R}$, take constant states $0, 1, \ldots$, all safe, and allow the edge $n \to n + 1$ on the open $U_n = (-1/(n + 1), 1/(n + 1))$. The piece of V_j at n is the intersection of the j successive opens from U_n. The meet over all j is empty, as the interior of their intersection is empty. Hence the ambient kernel is empty. At the stalk at 0, every edge is present, and the kernel is the full state set. This computes a strict comparison for geometric inverse image of an infinite-state kernel. The comparison has the direction $f^*K \leq K_{f^*}$, since the pulled-back postfixed certificate is postfixed by preservation of the predecessor formula.

CHAPTER 15

Controllers, games, and probabilities

15.1 Cycles and finite strategies

For a finite directed graph on V, a cycle means a nonempty closed directed walk. A lasso is a path from a specified state to a cycle together with that cycle.

Theorem 15.1 (Lasso characterization)**.** *The viability kernel of a finite graph is exactly the set of safe states from which a cycle of the induced safe graph is reachable. Every member has an infinite witness using a time-independent choice of outgoing edge for each visited state.*

Proof. An infinite safe path repeats a vertex after finitely many steps, so it contains a lasso. A lasso produces an infinite path by repeating the closed walk, so coinduction puts its source in K. A finite graph has finite branching, and Proposition 14.3 supplies an infinite path for every member. Alternatively, for each state of the kernel choose an outgoing edge into the kernel. This finite table is a time-independent controller and gives the last claim directly. □

Equivalently the safe graph's strongly connected components with at least two vertices or a self-loop contain the cycles, and the kernel is the backward reachability of those components. The reachability formula follows by traversing the prefix of a lasso. A backward search after computing the components therefore gives a linear-size graph certificate. Repeatedly deleting states with no surviving outgoing edge also gives the kernel, by the finite descending recursion.

15.2 An adversarial regime

Let a finite state x have a finite control set $U(x)$. For $u \in U(x)$, let $N(x, u)$ be the nonempty set of successors a disturbance can select after the control. Define

$$\Psi_{\mathrm{all}}(A) = \{x \in V \mid \exists u \in U(x),\ N(x, u) \subseteq A\}.$$

The order of play is part of the definition. A state with no admissible control lies outside the predecessor. The nonemptiness of the successor set means that a selected control always advances the game.

Theorem 15.2 (Finite safety game)**.** *The greatest fixed point of Ψ_{all} is the set of initial states from which a controller can keep every play safe. A fixed control table suffices on that set. From a state removed at a finite stage the disturbance forces failure within that stage's horizon, counting dead ends as failure to continue.*

Proof. At a fixed-point state choose a control whose successors stay inside the fixed point. Repetition yields safety and continued play for every disturbance. Let $V_0 = V$ and $V_{j+1} = \Psi_{\mathrm{all}}(V_j)$. For a state outside V_0, failure is immediate. At a state in $V_j \setminus V_{j+1}$, either no control is available or every control has a successor outside V_j. The disturbance chooses such a successor, and induction gives failure at the earlier horizon. A descending chain on the finite state set stabilizes, so every state outside the fixed point has one of these finite certificates. □

A model where the disturbance is observed before choosing a control uses the order $\forall d\, \exists u$. The ordering is visible in a two-channel repair. The disturbance chooses either channel 1 or channel 2 to fail. A controller restores one channel, and the transition stays safe exactly when it restores the channel selected. With disturbance first, choosing the corresponding repair gives a self-loop at the safe state. With control first, the other disturbance leaves safety. The two one-state kernels have different inhabitation.

15.3 Almost-sure safety

For a finite controlled stochastic model, take $p(y \mid x, u)$ as the transition probabilities. A safe successor support is $N(x, u) = \{y \mid p(y \mid x, u) > 0\}$. The robust support game of Theorem 15.2 computes the almost-sure safety region, even for controllers that choose randomized controls depending on past states.

Proposition 15.3 (Support semantics for probability one)**.** *A finite state admits a policy with probability one of infinite safe operation exactly when it lies in the greatest fixed point for the support game.*

Proof. A fixed safe control table keeps each step safe with probability one. The probability of an unsafe step at any finite time is zero, and a countable union of these events has probability zero. Conversely, from a state of finite removal rank, every control has an earlier-rank successor of positive probability or fails to continue. Under a randomized policy on finitely many controls, some chosen control has positive probability. Conditional on that control, there is a positive-probability successor of smaller rank. Continue recursively, using the history-dependent policy after that history. The ranks decrease, so there is a finite failure event of positive probability. Thus the probability of infinite safety is strictly below one. □

15.4 Finite-horizon risk

For $x \in V$, let $q_n(x)$ be the maximum probability of staying safe for the next n steps. Outside V put $q_n(x) = 0$. Let $q_0(x) = 1$ on V. With a dead end assigned zero probability of advancing, the recurrence is

$$q_{n+1}(x) = \max_{u \in U(x)} \sum_{y} p(y \mid x, u) q_n(y) \qquad (x \in V).$$

Conditioning on the first successor proves the upper bound. A maximizing control and independently maximizing continuation policies from the successors give equality. Finiteness supplies the maxima. By induction the condition $q_n(x) = 1$ is equivalent to membership in the n-th stage of the support game. Indeed a convex combination of numbers at most one equals one exactly when every positive-weight summand equals one.

A one-state safe region with a single surviving loop of probability $1 - p$ and a failure exit of probability p has $q_n = (1 - p)^n$. For $0 < p \leq 1$ the support-game kernel is empty. The existential graph kernel contains the state when $p < 1$, because a safe edge exists. These are two precise observations of the same controlled transition table.

CHAPTER 16

A common state for production and maintenance

16.1 The compiled transition

Let Z be a state object with coordinates for stock, energy, operating phase, and outstanding obligations. For a finite presentation, choose a family of permitted *round witnesses*. A round witness is a tuple of typed production trees together with an ordering of its primitive events. Its record includes a stock effect for each primitive event, an exergy charge, and a list of obligations discharged. The event order respects the dependencies of the trees. An environment schedule specifies arrivals on the same event clock.

A permitted event has a guard that checks stock, available exergy, phase, and other stated state predicates. When the guard holds, its successor is given by the stock and energy updates, the phase change, and the indicated discharge. An entire round also checks that all required obligations for that round were discharged and that the intermediate states satisfy safety. The object L of round transitions consists of an initial state, a round witness and arrivals, and the sequence of checked intermediate states. Its source and target are the first and final states. With deterministic updates the intermediate sequence is determined by the initial state, witness, and arrivals. A nondeterministic specification retains it as part of the witness.

Theorem 16.1 (Soundness and completeness of the round graph)**.** *Finite paths in the round graph are exactly the schedules of successive admitted rounds, with the same endpoints and the same labeled primitive events. In a finite round graph, a state admits indefinitely repeatable maintenance by admitted rounds exactly when it lies in the viability kernel.*

Proof. A single edge is exactly an admitted round by the definition of L. For successive edges, equality of intermediate endpoint states is exactly the compatibility needed to concatenate the rounds. Conversely a schedule of n successive rounds decomposes at its round boundaries, and each round satisfies the defining guards for one edge. Induction gives the finite path assertion. For a finite graph, Theorem 15.1 gives a reachable repeating cycle exactly from the kernel. Reading the labels of the cycle gives the repeated admitted maintenance. Conversely an indefinitely continued schedule gives an infinite safe path by the same concatenation, which yields membership in the kernel. □

This is a joint fixed-point construction. The least fixed point enumerates candidate production witnesses, indexed by budget and prerequisites. Admitted executions of those

witnesses form the graph used by the greatest fixed point. The same edge records the completed production, the discharge of obligations, and the next available state. The maintained property is tested on the resulting repeated schedules.

16.2 Two independent affirmative answers

Consider a state x with an idle loop and a repair operation of charge 2. The stock makes the repair's prerequisites present. Every interval supplies 1, the battery has capacity 1, and the required round includes one repair. The repair belongs to the structural production image. The idle graph has a viable state. In the round graph, every edge would include the repair, whose arrival-first guard requires a capacity of at least 2. Thus the round graph has empty kernel. The example assigns an exact charge and exact guards to the difference between finding a repair tree and actually maintaining a system with it.

A second example uses a charged repair that can execute once. Take one stock token m, an event $m \to r$ that uses it up, a constant ample exergy supply, and a round obligation r. The first round executes and leaves stock 0 in that coordinate. A second round requires another m, and no rule produces it. The round kernel is empty by deletion of the two states at the two horizons. Adding a second event $r \to m$ in the same round restores the stock. The initial state now has a loop and the kernel contains it. This isolates stock repeatability from the exergy test.

16.3 Round certificates that transport

Proposition 16.2 (Monotone admission at fixed state)**.** *Let $L \to L'$ preserve source and target and let the safe subobject of the first graph lie below the safe subobject of the second. Then $K_L \leq K_{L'}$.*

Proof. Every source image in the first predecessor is below the corresponding source image in the second. Therefore every postfixed certificate for the first graph is a postfixed certificate for the second. Taking the joins in Theorem 14.1 gives the comparison. □

For a map $f : S \to S'$ of distinct state spaces with every safe edge represented by a target safe edge, the image of K_L is below $K_{L'}$. To prove it, pull back the covering family of outgoing internal edges along the kernel, map those edges to the target, and take their source image. Their targets lie in $\mathrm{im}(f|_{K_L})$. Their sources cover that image by composition of effective epimorphisms. Thus the image is a postfixed subobject. A simulation of maintenance therefore transports a kernel witness forward. The invariant of a fully structured equivalence, as used in Chapter 18, transports both ways.

16.4 Progress conditions

A guard may require a particular completion at least once in each m steps. Add a counter $0, \ldots, m-1$ to the state, measuring the number of steps since its most recent

completion, with initialization at 0. An edge marked with completion sets the counter to 0. An unmarked edge increments it and is admitted only if the result remains below m. Induction on the steps shows that admitted paths have no run of m consecutive unmarked edges. Conversely, any path with that bounded recurrence admits the unique corresponding counter labels. Several bounded recurrence conditions use a product of these counters. This gives an exact safety compilation of recurring obligations and allows the same kernel procedure to check them.

For an unbounded requirement that a mark occur infinitely often on a finite graph, a state admits such a safe path exactly when it reaches a strongly connected component with a marked edge internal to it. For necessity, the vertices visited infinitely often by a path form a strongly connected set because they connect along the tail after the last visit to every transient vertex. Some marked edge recurs infinitely by finiteness. For sufficiency, a return path inside the component closes the marked edge into a walk. Reach it and repeat that walk. This is a different finite certificate for the unbounded recurrence specification.

CHAPTER 17

Periodic supply and exact battery thresholds

17.1 Two charge conventions

A period is a nonempty list $(a_i, c_i)_{i=0}^{m-1}$ of nonnegative supplies and charges, repeated cyclically. Write $d_i = c_i - a_i$ and $D = \sum_i d_i$. A battery of capacity C starts full at a specified phase. In a net ledger, the step is

$$e_{i+1} = \min(C, e_i - d_i), \qquad e_i - d_i \geq 0.$$

In an arrival-first ledger the step is

$$e_{i+1} = \min(C, e_i + a_i) - c_i, \qquad \min(C, e_i + a_i) \geq c_i.$$

The net ledger permits supply to meet current demand before the remaining balance is clipped. The arrival-first ledger clips the arrival before the charge. Both updates fit the balance in Chapter 12 with the corresponding spill.

For a circular block of consecutive indices use each index at most once. Define its sum using the d_i, and include the empty block of sum 0. Define

$$M = \max_{\text{circular blocks } I} \sum_{i \in I} d_i, \qquad M_{\mathrm{a}} = \max_i \left(c_i + \max_{0 \leq \ell < m} \sum_{j=1}^{\ell} d_{i+j \bmod m} \right).$$

Theorem 17.1 (Periodic battery theorem)**.** *Indefinite execution from a full battery is possible in the net ledger exactly when $D \leq 0$ and $C \geq M$. In the arrival-first ledger it is possible exactly when $D \leq 0$ and $C \geq M_{\mathrm{a}}$. These full-battery thresholds hold at every starting phase.*

Proof. In either ledger each completed period has charge minus supply D. If $D > 0$, the cumulative balance contradicts indefinite execution from a finite initial stock.

For the net ledger, unfold the update starting with $e_0 = C$, temporarily allowing negative values to record a failed guard. The result is

$$e_n = C - \max_{0 \leq h \leq n} \sum_{k=h}^{n-1} d_k.$$

For $n = 0$ the maximum is zero. At the next step the previous candidates acquire d_n and the clipping adds the empty suffix, proving the formula by induction. When $D \leq 0$, removing a full period from a suffix weakly increases its sum, so all suffix sums are

bounded above by M. Every circular block occurs along the infinite periodic word, hence M is attained as a candidate for some n. The guards all hold exactly when $C \geq M$.

For the arrival-first ledger, $e_{i+1} = \min(C - c_i, e_i - d_i)$. Unfolding gives, for $n \geq 1$,

$$e_n = \min\left(C - \sum_{k=0}^{n-1} d_k,\ \min_{0 \leq h < n}\left(C - c_h - \sum_{k=h+1}^{n-1} d_k\right)\right).$$

The first term is dominated by the $h = 0$ candidate because $c_0 \geq d_0$. In every other candidate one can remove full periods from the suffix when $D \leq 0$. Its resulting bound is one of the values in M_{a}. Every phase and following suffix of length below m occurs along the repeated word. Thus all e_n are nonnegative exactly when $C \geq M_{\mathrm{a}}$, and the guard is exactly this nonnegativity at each step. The maxima depend only on circular blocks, hence are independent of starting phase. □

17.2 Minimum initial stock at fixed capacity

For the net ledger set P_i to the maximum over the empty prefix and all prefixes of one period starting at phase i of their deficit sums. For the arrival-first ledger use the same P_i. It measures the term in the unfolded update that still depends on the initial battery. At capacity above the appropriate full-battery threshold, a starting charge e at phase i permits indefinite operation exactly when $e \geq P_i$.

For proof, replace the initial C in the unfolded first prefix term by e, and keep all clipping terms at capacity C. If $D \leq 0$, each long prefix has sum at most the maximum of those first-period prefixes, attained by one of them. A smaller e fails at that prefix. At $e \geq P_i$ all terms involving the initial charge are nonnegative, and the clipped terms are nonnegative by Theorem 17.1. The threshold is at most C because a full battery is feasible by that theorem. Thus the phase i contributes the entire interval $[P_i, C]$ of viable charges to the kernel.

17.3 Choice between maintenance modes

Consider a finite directed control graph with edge net deficits $d_e = c_e - a_e$. A control path takes exactly these increments. Start at a given safe vertex with freely chosen finite capacity and full initial charge.

Proposition 17.2 (Reachable cycle threshold)**.** *For either ledger, some finite capacity sustains an infinite path exactly when a directed cycle of nonpositive total deficit is reachable in the safe graph.*

Proof. If a cycle has nonpositive deficit, select a finite path to it and then repeat the cycle. The deficit sums of all blocks in that concatenation are bounded above. Blocks within the periodic tail have the bound of Theorem 17.1, and a block touching the prefix adds only a fixed finite cost. In the arrival-first ledger each leading charge is also bounded

on the finite edge catalog. The same unfolded update as in Theorem 17.1 therefore permits a sufficiently large capacity.

When every reachable simple cycle has positive deficit, the following path bound applies. In any path remove simple closed subpaths until a path with no repeat remains. The remainder has length bounded by the number of vertices. If reachable cycles exist, their positive mean deficits have a positive minimum δ, since there are finitely many simple cycles. The removed parts of a path of length n have total deficit at least $\delta(n - |V|)$. The residual deficit is bounded below by a fixed finite constant using the bound on its length. Thus the cumulative net deficit grows without bound, contradicting the cumulative balance for a fixed finite initial charge. If there are no reachable cycles, all path lengths are already bounded by the number of vertices. □

The choice theorem supplies an existence test for capacity as a design variable. The exact optimum at fixed capacity is computed by forming the lifted graph with stock coordinate and applying Theorem 15.1. For integer charges, capacities, and supplies the lifted graph is finite. The two formulations share the certificate given by a reachable lasso in the lifted graph. This lasso projects to an admissible sequence of choices in the base graph, with its quantitative state carried by the lift.

CHAPTER 18

The invariant of a bounded system

18.1 The structured record

A bounded system over B consists of the following data.

1. A material catalog and a component typing $C_X \to K_{\mathrm{mat}}$.
2. An organizational fact object K_X, a demand map $D_X \to K_X$, and an endogenous rule presentation on K_X, with a physical interpretation of its operation profiles.
3. Admissible boundary inputs, grade maps, and a stock-and-ledger regime that specifies a state object S_X, safety subobject V_X, and a graph L_X of admitted rounds.
4. A typed catalog of supplements, with a compatible interface and interpretation for each regime in which the supplement is joined to the system.
5. An observation of its computational behavior, when that datum is used for comparison.

Each round supplies its witness tuple, including the organizational targets it discharges. The stock-and-ledger regime specifies which demands recur together, and which initial states are eligible. This last datum is a subobject $I_X \hookrightarrow S_X$. The comparison of viable kernels uses $K_X^{\mathrm{run}} \wedge I_X$ when initial eligibility is part of the judgment.

Define G_X as the image of all endogenous trees, A_X as the image of demand, and Q_X as the greatest fixed point on its round graph. Write

$$\mathcal{J}(X) = (\mathrm{Supp}(X), A_X, G_X, \mathrm{Att}_X, V_X, Q_X, I_X \wedge Q_X)\,.$$

Here Att_X is the attainable-budget relation for the endogenous trees, and all entries retain their types. The same invariant can be evaluated in each named coupled regime. An equivalence of bounded systems is a coherent equivalence of the displayed data, including compatibility of interpretations and of admitted events. Equivalences are compared at the stated boundary and scale.

Theorem 18.1 (Invariance of the organizational record)**.** *An equivalence of bounded systems transports $\mathcal{J}$ entry by entry. It transports the corresponding record in every paired supplement regime. Closure and local inhabitation of the eligible viability kernel are invariant.*

Proof. The image construction transports support and demand. The equivalence acts on the finite tree diagrams and on the grade and budget comparison. Their images transport G_X and Att_X. The predecessor formulas are conjugate under the induced isomorphism on state subobjects, so the greatest fixed points agree by the transport

argument of Chapter 4. The initial-state subobject and the safe subobject are preserved by definition of equivalence, and intersection and context images transport. For a paired supplement, equivalent diagrams of operads have equivalent homotopy pushouts by their mapping property. The coupled interpretations and admitted round graphs are then compared by the same arguments. □

18.2 Structural and sustained autonomy

Structural closure is the judgment $A_X \leq G_X$. Sustained endogenous maintenance is structural closure together with $\mathrm{im}(I_X \wedge Q_X \to B) = B$. At a point this asks for an eligible state with a viable round schedule. In a demand mode whose repair is compulsory in every admitted round, any such eligible state supplies a witness of that repair. The proof is that it has an outgoing kernel edge locally, and the label on that edge contains the required tree. The eligible context image being full means these local witnesses cover the context. In particular a static proof of closure and the state certificate can be checked on the same data.

For a supplement H define $\mathcal{J}_H(X)$ by coupling along its typed interface. In a finite set model, a *dependence certificate* consists of a demanded role in $A_X \setminus G_X$, a coupled production certificate for all demands, and a viable start in the coupled regime. More generally the shortage witness is a section locally through $A_X \wedge \neg G_X$ with a specified context, and the positive claims are checked by effective descent. A *power dependence* has the full endogenous production image but an empty eligible kernel at one supply regime and an inhabited eligible kernel at a stated larger regime. These two certificates record different changes and can be combined.

18.3 Named classes of supplements

A biological host is a label on a supplement regime that includes a host's contributions at a specified interface. An industrial environment is a label on one that includes the resources and transformations of a fabrication chain. The labels are part of a typed catalog. Relative closure with either label has the same order-theoretic expression. A system can carry several alternative supplements. Its record is then a function on regimes, not a single Boolean tag.

Proposition 18.2 (Regime comparison)**.** *With a fixed demand family and fixed state object, inclusion of sources, rules, and admitted transitions yields an order comparison of generated images and viable kernels. If the only added input is a larger budget for the same witness family, the ungraded generated image is unchanged and the attainable slice enlarges.*

Proof. Inclusions of rules and seeds transport trees and their images by the construction in Chapter 5. Inclusion of admitted transitions transports postfixed certificates by Proposition 16.2. A change of budget for a fixed presentation selects a larger subfamily

of witnesses by Proposition 10.3. The ungraded image uses their union over the entire rule presentation. □

18.4 Two exact separation criteria

Theorem 18.3 (Witnessed separation)**.** *Two bounded systems over a point with the same compared catalogs belong to distinct equivalence classes whenever one of their material supports, generated images, attained-budget relations, or viable-state context images differs under every identification of the typed catalogs that is allowed by the record. In particular, an autonomous system and one with a deficit at the same organizational signature lie in distinct equivalence classes.*

Proof. Every equivalence induces one of the allowed identifications and carries all four entries to their counterparts by Theorem 18.1. In the last case, closure is preserved under every such identification. Transporting the inclusion $A_X \leq G_X$ would supply closure for the other system, in contradiction to its deficit witness. □

Proposition 18.4 (Four independently realized outcomes)**.** *For finite systems with one repair role, both structural closure and inhabitation of the safe operating graph can have either Boolean value, independently. For the round graph with the repair compulsory on every edge, viable operation implies structural closure.*

Proof. For the independent pair of coordinates use one organizational fact, with or without a production seed, and independently take a singleton safe state with or without an idle loop. These four presentations realize the four pairs. For the round graph, a viable state has an outgoing admitted round and its label supplies the compulsory repair. This proves structural closure for the one repair role. The same proof applies to any demand family included as obligations in every round. □

CHAPTER 19

Observations and physical equivalences

19.1 A category of observations

An observation is a functor from structured systems to an observation category. An input-output observation can assign the transition traces of a declared interface, leaving other labels out of its codomain. A dependency observation can assign a graph of which declared subsystems supply which others. At a finer resolution the observation includes the entire production presentation. These are distinct choices of data to transport.

For an exact example, use a category whose objects are pairs of a demand diagram and a finite deterministic input-output automaton. A map consists of a map of diagrams and an automaton morphism. The projection to automata forgets all organizational data. Fix the automaton with one state and one output, and compare the two diagrams with $R = D = \{r\}$, $P = \varnothing$ and $P' = \{r\}$. There is a map over the identity of the automaton given by the unique inclusion on production objects. Its image is an identity, while the source map has distinct production cardinalities and so is outside the equivalences. The functor is nonconservative. The two objects have identical traces for every input history and different closure judgments.

19.2 Proof of physical distinction with matching traces

Theorem 19.1 (A common computation at three boundaries)**.** *For any fixed automaton M, three bounded presentations can share its complete input-output behavior while differing in autonomous maintenance, host-relative maintenance, and material support with industrial maintenance.*

Proof. Use the product with the same automaton observation in each case. Take one repair obligation r, a battery of capacity 1, and a singleton operating phase with supply 1. In the first system the charge-one repair is endogenous, so the full-battery state executes a loop and the required fact is generated. In the second system the repair appears only in the coupled catalog of a named host. The endogenous generated image omits r, and the host coupling restores the loop. In the third system the charge-one repair appears only in a named industrial coupling. Give that system a different support in the material catalog. All comparisons of repairs are made at the one demand label. The first system is distinguished from each dependent system by Theorem 18.3. Support distinguishes the two dependent systems. The observation in each case is M. □

The same construction works with a computation described by an arbitrary transition system in place of a finite automaton. In each case the computation is the observation datum carried along unchanged by the organizational and ledger comparison. The example establishes a precise independence of trace semantics and physical repair data.

19.3 A localization of observations

In the sheaf setting, restriction to an open U is an inverse image that changes the observed context. A sheaf supported outside U restricts to the initial sheaf, so outside data can distinguish systems with identical observations on U. For an explicit model take the disjoint union $B = B_1 \amalg B_2$ of two points. Use a common singleton demand over both points. One production covers it at both points and one only at B_1. On B_1 the two production diagrams are identical. The closure loci on B are B and B_1. Their distinctness is preserved by every effective cover of B, by image descent. Restriction to B_1 computes a localized equivalence in the category of observed diagrams.

More generally, a left-exact localization carries an attained tree-image construction to its localized image by Corollary 5.4. It carries a finite bounded kernel to its localized kernel by Theorem 14.4. Thus on a uniform finite model the closure and viability comparisons can be made internally in the localized observation category. The loci are recorded in that category, and the comparison on the ambient catalog uses the underlying full diagrams. This makes an observation-dependent notion of viability available without changing the definition of a physical equivalence.

19.4 Simulation of states

For graphs S, S', a simulation relation $R \subseteq S \times S'$ relates states and specifies that each transition from a related first state has a matching transition from the second, with related endpoints. Induction on the length of a path gives a matching finite trace. If a simulation also preserves safety and control labels, it transfers safe traces at that resolution. A pair of simulations can yield the same trace language without an isomorphism of graphs. Take a one-state loop and two disjoint copies of it, with all labels equal and all states initial. Both have exactly the traces consisting of repeated copies of that label. The relation connecting the singleton to both copies is a simulation in both directions. Their state objects have distinct cardinalities. This distinction persists when the state graph is retained as part of a bounded system.

The canonical graph map from the two copies to the singleton sends every safe edge to a safe edge. Its viable-state image is the singleton by the simulation comparison in Chapter 16. Recording only the context image of the kernel is therefore a further observation of the full invariant. In sheaves, replacing the two globally separate copies by the torsor of Example 2.3 gives the same local context image while changing the global sections of the kernel.

CHAPTER 20

A regenerative cellular model

20.1 Roles and boundary

The cellular archetype maintains a boundary transporter b, a conversion machine m, and a copying module g. Feedstock f arrives at the boundary. The organizational demand records replacement of the transporter, conversion machine, and copying module. We denote the three completed replacements by $\widehat{b}, \widehat{m}, \widehat{g}$. The same type of distinction between a material flux and a constraining process appears in the biological closure literature. In the chemoton model, metabolism, membrane, and genetic replication are coupled components of the organization. [18, 19, 23]

Use a stock of one b, one m, and one g, and the following period of four events. A catalyst is recorded on both sides of its stock rule.

event	input	output	charge
t	$f + b$	$p + b$	1
c	$p + m$	$q + m$	2
w	$q + g$	$b' + m' + g' + g$	3
r	$b' + m' + g' + b + m + g$	$b + m + g$	1

Here r retires the old machinery and installs the primed replacements, relabeling them as the next round's b, m, g. The total output at w packages one replacement of each role. Their discharge is recorded by r. The material ledger admits one input f at the start of every round, and removes the retired parts through the boundary at r. Charge and stock are independent coordinates of the graph.

Proposition 20.1 (Exact cellular round)**.** *The cellular model supplies all three organizational demands by endogenous trees and restores its initial machinery stock in every admitted round. With exergy arrivals* $(2, 2, 2, 1)$ *in the four phases and the arrival-first convention, its minimum full battery is* 4*.*

Proof. The Boolean projection produces p, then q, then each primed part, and finally the three completion labels. The trees have stages $1, 2, 3, 4$ respectively. Stock execution starting with the three units and an arriving f enables those same events, and the replacement event restores one unit of each machinery role. The charges are $(1, 2, 3, 1)$, the deficits are $(-1, 0, 1, 0)$, and their sum is zero. The four leading-charge expressions in cyclic order are $2, 3, 3, 1$, whose maximum is 3. For a direct calculation at capacity 3, the full start follows the energy sequence $3, 2, 1, 0, 0$, and the next start with charge 0 follows $0, 1, 1, 0, 0$, which repeats. Capacity 2 fails the charge-3 guard. □

20.2 A pulse arriving at the boundary

Shift the supply to $(4, 0, 0, 3)$ with the same total seven. The deficits become $(-3, 2, 3, -2)$, with total zero. For the leading charge at phase 0, the two-step suffix after it has sum 5, so $M_{\mathrm{a}} \geq 6$. The four leading-charge candidates are $6, 5, 3, 3$, giving $M_{\mathrm{a}} = 6$. At capacity 6 an initial full stock follows $6, 5, 3, 0, 2$, after which the state of charge 2 begins the recurrent energy round $2, 5, 3, 0, 2$. The primed machinery still passes through the same stock transitions. The altered environmental rhythm changes viability through the battery coordinate while the endogenous productions remain the same.

20.3 Distinguishing energy delivery from repair delivery

Consider two alternative changes. The first replaces exergy arrival at a phase by a larger quantity. It changes only the ledger guard. The second replaces event w by delivery of the three primed modules, represented as a supplement operation. It changes the proof family. The endogenous repair witness using w is marked by an internal vertex of conversion. The delivery witness is marked by the supplement. If w is removed, b', m', g' form a missing-set certificate for the endogenous presentation. Delivery invalidates that certificate by adding a producing rule with no prerequisite in the marked missing set. Extra exergy leaves the certificate valid because the endogenous rule list is the same.

20.4 Local coordination across scales

The model can be indexed by a context B of environments. At each point, availability of f is a leaf condition and the battery schedule is a function of the context. In slices with discrete finite state and integer capacity, one obtains the viability locus by a finite safe-path image. If the feedstock condition holds only over an open subobject of B, the raw transport edges are defined there. The viability kernel lies above that open, since every admitted round requires a transport. When the supplies on the open meet the full-battery threshold and the initial state contains the required stock, the eligible kernel image equals that open. Necessity uses the transport guard. Sufficiency is the explicit cycle transported over the open.

CHAPTER 21

A host-dependent replicator

21.1 Separating replicator and host roles

The viral archetype uses a replicator token v and external machinery for translation and copying. A host role h converts supplied precursor f into an available helper m. The replicator's program supplies the operation $a : (v, m) \to p$ that prepares a precursor for a descendant, and $b : (v, p) \to v'$ completes its assembly. We take demand to be the occurrence of a completed descendant $\widehat{v}$, not the presence of the original token v. Co-opted host factors and the use of host protein synthesis motivate the split between helper production and replicator-controlled operations. The infected cell and the free virion occupy different boundaries in this modeling vocabulary. [26, 27, 25]

Theorem 21.1 (A host creates the missing proof)**.** *With seed v and no seed m, p, v', the set $\{m, p, v', \widehat{v}\}$ is a missing-set certificate for the replicator. With host production $h : f \to m$ and an admitted seed f, the coupled operad has a mixed witness for descendant assembly and the coupled fact presentation derives $\widehat{v}$.*

Proof. Every endogenous producer of p needs m, every producer of v' needs p, and the completion rule needs v'. No endogenous rule produces m. All marked facts are disjoint from the seeds, so the missing-set criterion applies. In the coupled term, substitute h into the helper input of a, then substitute that result into the precursor input of b. These are well-typed mixed substitutions in the pushout. With the two seeds, its evaluation has root v', and the completion rule yields $\widehat{v}$. □

21.2 A repeatable infected-cell round

Take the three events to have charges $1, 2, 3$. Let the host round produce one helper using a fresh feedstock unit, let preparation consume the helper, and let assembly return the persistent replicator and emit one descendant. Descendants leave through a separate counted output. The stock at the start of the next round is therefore the same as the stock at the start of the first, and each execution produces one descendant. Put environmental arrivals $(0, 4, 2)$ on the event clock. The deficits are $(1, -2, 1)$, with sum zero. The leading-charge candidates for the arrival-first convention are $1, 4, 4$, so the full-battery threshold is 4. At capacity 4 the initial energy path $4, 3, 2, 1$ reaches the periodic energy path $1, 0, 2, 1$. These paths execute the mixed proof under the common ledger.

This calculation also classifies the regime obtained by deleting the host machinery while retaining the same energy trace. The first event can deliver its energy to the boundary, but no rule in the reduced presentation produces m. The missing set from Theorem 21.1

proves that every permitted complete round has a missing witness. The round graph has no completing edge. Restoring host machinery with a total arrival strictly below six per period supplies the proof and still gives empty viability by Theorem 12.2. These two changes have distinct invariants.

21.3 Multiple host pathways

Suppose the replicator accepts either helper m_1 or m_2. Host H_1 provides a producer for m_1 at one grade, and H_2 provides a producer for m_2 at another. Then the minimal source portfolios for a descendant are the two singleton host sets, as each appears in one tree and every tree contains one helper-producing vertex. If the replicator has two distinct helper prerequisites, with the first supplied only by H_1 and the second only by H_2, the minimal portfolio is $\{H_1, H_2\}$. This follows from the leaf-source characterization in Chapter 5. Adding an endogenous alternative for one helper replaces the joint requirement by the remaining singleton. The exact portfolio records the architecture of dependence beyond the truth value of the single host-relative closure query.

21.4 A typed host switch

Let x record the chosen host environment. Suppose a round at $x = 1$ admits only the path through m_1, and a round at $x = 2$ only the path through m_2. If the host coordinate alternates, indefinitely successful operation requires both paths to be executable on their own phase. The product graph makes this condition a two-phase cycle. If the two phase paths have equal stock effects and there are separate arrivals, their joint energy threshold is obtained by concatenating their event schedules and applying Theorem 17.1. A single cheapest descendant proof gives one of the two production charges, while the alternating kernel uses both witnesses. This example gives an exact operational difference between existence of a host and a prescribed succession of hosts.

CHAPTER 22

Computing machinery and fabrication

22.1 Two signatures of a computer

A computational signature may ask for update, memory, and output operations. A physical-maintenance signature also asks for restoration of the processor, replacement of damaged memory, and assembly. A machine executing a fixed program has an automaton or a richer transition system at its observational interface. We attach a distinct physical presentation for repair. This separation of descriptions supports the distinction between a device and the industrial chain that builds and maintains its components.

Use facts s for running software, d for damaged hardware, z for a replacement chip, and a for a repaired assembly. The device has a computational operation $s \to s$ and a local installation operation $(d, z) \to a$. It starts with s, d. The physical demand is completion $\widehat{a}$ after installation. Without a source of z, the set $\{z, a, \widehat{a}\}$ is a missing-set certificate. The proof is direct inspection of the only rule for assembly and the completion rule. A fabrication supplement producing z invalidates the certificate and supplies a mixed derivation. This mechanism separates physical reproduction or repair from execution of the installed program.

22.2 A fabrication boundary tower

Consider four regimes ordered by adjoining the successive catalogs

$$\begin{aligned} D_1 &= D_0 + \text{power}, \\ D_2 &= D_1 + \text{installation supply}, \\ D_3 &= D_2 + \text{fabrication and recycling}. \end{aligned}$$

At D_0 and D_1 the replacement chip has no producer. At D_2 a delivered replacement supplies z. At D_3 a rule from material precursor to z provides a second proof. If the delivered replacement is a finite stock of n tokens, each charged installation consumes one. The physical-round kernel at D_2 is empty when the installation is compulsory in every round, by the rank given by the number of remaining tokens. At D_3, a fabrication rule replenishes the token and can close a recurrent stock cycle. The fabrication event and the installation event together give the round proof.

Proposition 22.1 (A recurrent industrial schedule)**.** *Let the fabrication and installation charges be* 5 *and* 2*, and let their respective arrivals be* 6 *and* 1*. With one fresh precursor per round, a persistent base assembly, and installation consuming the fabricated token, the*

minimum full battery is 6 *in the arrival-first ledger. The physical-round kernel contains a recurrent assembly state.*

Proof. The stock returns to the same start, with the freshly produced token consumed in installation. The deficits are $(-1, 1)$, so $D = 0$. The candidate at fabrication is $5 + 1 = 6$, and the candidate at installation is 2. Thus the threshold is 6. At that capacity the energy path $6, 1, 0$ reaches the periodic path $0, 1, 0$, because each cycle's first arrival supplies the charge 5. The stock and energy paths together give a recurrent round. □

22.3 An artificial regenerative system

A self-fabricating device archetype replaces the external chip producer by an internal rule g whose prerequisites are an admitted raw precursor and persistent fabrication machinery. Add an internal maintenance rule restoring that machinery, with a dedicated completion label for its own replacement. A round sequences g, hardware installation, and fabrication-machine restoration. With charges $(5, 2, 3)$ and arrivals $(5, 2, 3)$, the deficits are all zero. The arrival-first threshold is the maximum charge 5. If the stock updates restore one copy of both machines and import only the declared feedstock, every phase has an admitted witness and the round closes a cycle. This is an explicit inorganic model of the same invariant as the cellular archetype.

22.4 Pairwise separation at specified regimes

Theorem 22.2 (Cell, viral unit, and device)**.** *Equip the cellular archetype with the endogenous round of Chapter 20, the replicator with the isolated and host-coupled regimes of Chapter 21, and the conventional device with the isolated and industrial regimes above. Give the two biological presentations material types in a catalog with a distinct device material type. The three isolated records belong to distinct physical equivalence classes. Each dependent record has an inhabited maintenance kernel at its specified enlarged boundary.*

Proof. The cellular archetype closes its entire organizational demand and has a viable endogenous round. The isolated replicator and isolated device have the explicit missing-set certificates above, so closure separates each from the cellular archetype. The support of the device differs from that of the replicator, so Theorem 18.1 separates that pair. The host schedule in Chapter 21 gives an eligible repeating energy and stock state, and Proposition 22.1 gives one for the device after industrial coupling. □

There is also a common dependency observation of the two enlarged systems. Send each pair of boundaries to the directed graph with one distinguished dependent vertex and one supplying vertex, with a supply arrow from the second to the first. Both pairs map to the same graph with that typing. The material record and the mixed proof

profiles determine the finer comparison. This realizes the analogy of external dependence at an explicitly specified level of observation.

22.5 Delivered inventory and recurrent fabrication

Vary the number n of delivered units while keeping the installation rule fixed. Let x count the number of delivered units left before a required installation. With ample power and an otherwise persistent assembly, the projected round graph is the chain

$$n \longrightarrow n-1 \longrightarrow \cdots \longrightarrow 1 \longrightarrow 0.$$

The last state has no completing round. Inductively, state j permits exactly j further rounds and has deletion rank $j+1$. The Boolean catalog already supplies z for every $n \geq 1$. This family gives arbitrarily long successful prefixes under a single affirmative production judgment.

Adding fabrication replaces the decrement by a two-event round that first creates one token and then consumes it. The next round sees the same x. At sufficient power its lifted graph contains the cycle calculated in Proposition 22.1. The stock projection therefore has an identity effect at every start that admits the new event sequence. The renewable input takes the form of material for fabricating a part.

CHAPTER 23

Heredity, export, and repeated birth

23.1 A productive event with a typed descendant

Add a type catalog G of heritable states to the operational model. A birth event records the parental type g, the offspring type g', and the repair and fabrication witnesses used for producing the offspring. A retention relation $R \hookrightarrow G \times G$ specifies the inherited transitions to be counted. The born unit can leave the maintenance boundary, in which case a separate export counter records its production. The parent viability graph retains the parent state after that export.

If types are spaces, an offspring correspondence has maps to G for the parent and offspring, and retention is enforced by pullback along R. Any symmetry of the type catalog that transports the correspondence also transports its image on $G \times G$. In a higher topos, that is the same image invariance as the invariance of physical support. The correspondence retains the space of realized parent-offspring pairs, while its image records their locally possible pairs.

Proposition 23.1 (Exact lineage paths)**.** *In a finite presentation with one maintained parent unit at each step of a lineage, the image of n-fold pullback composition of its admitted birth relation is the relation of type pairs joined by n admitted births, when the vertices of the relation retain the full state at which the next birth begins.*

Proof. Composition is an image of the fiber product over the intermediate vertex. Its elements are precisely pairs of composable birth witnesses with the same intermediate state. Iterating this description yields strings of n witnesses. Projecting to the type pair at the endpoints gives exactly the asserted image. Composition retains all state coordinates at the gluing interface. □

The full state at a birth boundary can include the stock of inherited catalysts and the charge of the initial energy store. Projecting a birth to types and then composing those projected edges allows intermediate matches with different stocks. For example, a birth from type g to type g can consume the one available catalyst. Its type projection has a self-loop, and the state relation has an edge from positive to zero catalyst stock. Composing the actual state edges gives no two successive births. This is the lineage counterpart of the round compilation.

23.2 A recurrent export condition

Mark transitions that export a viable offspring. In a finite safe graph, existence of an infinite parent history exporting infinitely many offspring is equivalent to reachability

of a safe strongly connected component with an internal marked edge. The argument in Chapter 16 supplies the exact criterion. If every period of a lasso contains b such marked events and the period has ℓ transitions, its asymptotic export rate is b/ℓ. The finite prefix disappears when the number of exports is divided by elapsed transitions, and the incomplete terminal period contributes a bounded error.

Theorem 23.2 (Maximum repeated export rate)**.** *Give each edge of a finite safe graph a nonnegative export count. From a start that reaches a cycle, the supremum of upper asymptotic export rates over infinite histories is the maximum reachable mean export count of a simple directed cycle. A lasso attains it.*

Proof. Let M be the maximum reachable cycle mean. Decompose any finite path by removing simple cycles until a simple path remains. The removed counts are at most M times the number of removed edges. The remainder has bounded length and bounded export count uniformly over paths. Dividing by the original length and taking the upper limit gives an upper bound of M. A path to a maximizing cycle followed by indefinite repetition has rate M, as the prefix and incomplete cycle contribute bounded errors. □

23.3 A two-type population arithmetic

Suppose a counted round produces a deterministic number of offspring of each type, and each new individual in turn starts a round at its own admissible initial state. Let M_{ij} count offspring of type j from a parent of type i. The row vector of generation counts satisfies $n_{t+1} = n_t M$, and hence $n_t = n_0 M^t$. This follows by grouping all children by their parental type and summing the counts. The counted generation model treats each replicated unit as running its own admitted round with its allocated inputs.

For $M = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$ and $n_0 = (1, 0)$, induction gives $n_t = (F_{t+1}, F_t)$ for the Fibonacci sequence $F_0 = 0, F_1 = 1$. This is a population-level computation of type counts, derived after the operational viability of the individual rounds. If the only admissible type-2 state has no completing round, then the second row is zero when only admitted birth events are counted. The resulting type counts are $(1, 1)$ at every positive generation. Both matrices describe the same initial type and different admitted lineage data.

23.4 Invariance under change of inherited labels

A bijection of the type catalog that preserves the birth correspondence conjugates the count matrix by the corresponding permutation matrix. Induction on t yields the same relation for its powers, and the two count vectors are related by the relabeling at every generation. In an internal model, an equivalence of the full type and birth diagrams transports the iterated pullback relations before taking their images, so all finite-horizon lineage comparisons transport as well. The construction thereby extends the invariance theorem from the continued maintenance of one unit to the typed relations among

descendants.

23.5 Budgeted lineage counts

Let a birth path have a grade in an ordered commutative monoid and a source type and target type. When the parent and offspring operational states match, sequential composition adds the grades. Projecting the graded relation to the types gives an upper set of budgets for each pair and each specified number of generations. A lineage of length n with grade at most b supplies a point of that upper set. Conversely its membership arises from an actual path by the same image definition. In a finite model this is a sequence optimization over the full boundary states. Taking the Pareto reduction of grades after each step preserves exactly the upward closure of possible grade vectors, since all later charges add monotonically.

A population constrained by a common finite budget needs a further state coordinate for the shared allocation. A two-type count matrix gives potential offspring counts per admitted parent round. To realize those counts together, one can order the parent rounds, debit each charge from the common store, and apply the ledger guard at each debit. This is a single finite composite action for one generation. Its cumulative cost bound follows by summing the prescribed rounds, and its viable recurrence follows by keeping the shared ledger in the state graph. The construction distinguishes the ability of each unit to reproduce from the throughput available to the whole collection.

CHAPTER 24

Design and minimal supplementary capacity

24.1 A partially ordered design space

A design specifies a portfolio of typed supplements, a supply regime, a battery bound, and a collection of permitted initial states. Fix a finite presentation and call a design successful when its required demands are produced and its initial-state set meets its round kernel. Define an enlargement to add seeds or rules, keep existing admitted rounds, and enlarge the eligible region. Proposition 18.2 shows that the successful designs form an upper set. For scalar capacity with fixed arrival-first rules and no energy-dependent guard except affordability, the map $(x, e) \mapsto (x, e)$ identifies each smaller admissible run with a run at greater capacity using at least as much energy at every stage: induct on the monotone update $e \mapsto \min(C, e + a) - c$. When safety has no upper charge guard except capacity and the other state coordinates agree, the corresponding successful capacity set is also an upper set. Full starts at the two capacities satisfy the required initial comparison.

In a finite parameter grid, every nonempty upper set is determined by its minimal elements. For any successful element, descend within the successful lower interval until it is minimal. Conversely every element above a minimal successful one is successful. This justifies keeping a frontier of minimal designs. When portfolios and batteries are both variable, the frontier records whether adding a source reduces the required battery by introducing a lower-charge proof or by changing timing.

24.2 A linear certificate for obligations

Let event e discharge a nonnegative count vector a_e and have charge c_e. Take a nonnegative vector of shadow prices λ with $\lambda \cdot a_e \le c_e$ for every permitted event. Every history with total obligation count at least b then has total charge at least $\lambda \cdot b$. In proof, sum the inequalities over the history and use monotonicity of the dot product with $\lambda \ge 0$. This gives a certificate that a proposed cumulative supply fails when it is below that amount, even for events discharging multiple roles together.

For two obligations and events discharging $(1, 0), (0, 1), (1, 1)$ at charges $3, 4, 5$, the price vector $(2, 3)$ is feasible. A round requiring one of each costs at least 5, and the joint event realizes the bound. The price calculation prices the actual discharge vectors rather than summing minima for the two roles independently. At a throughput of one round per interval, this example realizes the exact bound 5 in every interval before accounting

for timing.

24.3 Critical sources and certified alternatives

For a target with minimal portfolios $\mathcal{M}$, a source i is present in every successful portfolio exactly when $i \in \bigcap \mathcal{M}$. A deletion set F destroys availability exactly when F meets every $M \in \mathcal{M}$. The remaining source set contains one of the M's precisely when that M is disjoint from F. Hence minimal failure sets are the minimal transversals of the minimal-success hypergraph. The condition F meets each M is a finite verification of architectural vulnerability.

For multiple simultaneous targets, take the family of unions of one portfolio from each target, and reduce it by inclusion. Each union provides all targets and every successful portfolio contains one such union. At finite budget, include the corresponding grade sums if proofs are charged independently, or include joint discharge rules when sharing occurs. This gives exact substitutions of separate demands into a joint portfolio calculation.

24.4 An explicit capacity tradeoff

Suppose a device can use a direct repair e_1 of charge 6 with one incoming pulse of 6, or a supplemented two-step route with charges $(3, 3)$ and arrivals $(3, 3)$. At the original boundary the full-battery threshold is 6. With the added source, capacity 3 suffices on the second route. If that source consumes a fixed external allocation of two units outside the local ledger, the local capacity frontier still contains $(\varnothing, 6)$ and $(\{H\}, 3)$. The combined-system throughput must include those two units where its boundary includes their cost. This follows by the ledger sum of Chapter 12. Local storage and combined expenditure give two different coordinates of a design.

24.5 Finite certificates in both directions

To establish success, exhibit a tree for each demand and a path from an eligible state to a safe cycle in the round graph. To establish failure, a missing-set certificate proves a structural shortage. When the production test succeeds, list the deletion rank of each eligible state, with an earlier rank for every possible safe successor. These ranks prove absence of an infinite safe history by the well-foundedness of the natural numbers. The kernel algorithm in the appendix yields both the successful cycles and the finite deletion ranks. Thus the frontier of successful designs is supported by both production evidence and a separate operational certificate at each tested boundary.

CHAPTER 25

Context-dependent regulation

25.1 A finite context site

A finite poset P can index refinements of an observational context. Use its Alexandrov topology, with open sets the upward-closed subsets. The smallest neighborhood of p is $\uparrow p = \{q : p \leq q\}$. Subterminal sheaves identify with open sets, with intersection as meet, union as join, and the empty set as bottom. A constant finite family of capabilities K gives one copy of this frame in each capability. A subobject of the constant family is specified by opens U_k on which capability k is available.

For opens A, E the implication is

$$A \Rightarrow E = \{p \in P \mid \uparrow p \cap A \subseteq E\}.$$

The right-hand set is upward closed. Its intersection with A lies in E because $p \in \uparrow p$. Any open W with $W \cap A \subseteq E$ lies in that set, since $\uparrow p \subseteq W$ whenever $p \in W$. This proves the implication formula. In particular, the set-theoretic region where a point fails its current observation gives a closure locus after deleting all its predecessors as well.

Proposition 25.1 (Finite locus calculation)**.** *For finite constant capabilities K, let A_k, E_k be the open demand and endogenous availability for k. Write $F = \bigcup_k (A_k \setminus E_k)$, considered as a subset of P. The closure locus is $P \setminus \downarrow F$. After adding open supplemental supply H_k, it is $P \setminus \downarrow \bigcup_k (A_k \setminus (E_k \cup H_k))$.*

Proof. Proposition 2.4 takes the meet of $A_k \Rightarrow E_k$ over the fibers of the constant catalog. By the implication formula, a point belongs to this meet exactly when none of its refinements belongs to F. This is precisely the complement of the downward closure of F. Supplementation replaces E_k by its join with H_k, which is the union of opens. □

25.2 Local fixed points from a varying rule table

Give each seed and rule an open of availability. The stagewise formulas of Chapter 5 use unions across alternatives and intersections across prerequisites. At a point p, set membership in these unions and finite intersections is evaluated in the usual Boolean way. Therefore the available facts at stage j at p are exactly the stage-j facts of the rule table formed by retaining only seeds and rules active at p. Induction on j proves the claim. The union over the stages gives the pointwise fixed-point facts. In a finite table the uniform rank bound of Proposition 5.5 gives a finite procedure for every point.

For a finite state graph, let each safe state and each edge have an open on which it is present. The finite predecessor formulas similarly use unions and finite intersections.

By Theorem 14.4, evaluating the kernel at p equals computing the kernel of the finite graph seen at p. An initial-state open further intersects the kernel open at each state, and taking the union gives the open of eligible viability. Both the rule calculation and the kernel calculation preserve upward closure because the tables at a refinement contain the tables at its predecessor. Demand can enlarge along a refinement, so the locus of demand fulfillment uses the implication formula as a separate step.

Example 25.2 (A branching context)**.** Let P have a bottom vertex o and two maximal refinements a, b. A demand for repair is active everywhere, endogenous supply is active at a, and a host repair is active at b. Endogenous closure holds on $\{a\}$. Coupled closure holds on $\{a, b\}$, whose union is still proper because the root is outside it. If a safety loop is available at a and a completing host loop at b, the same open $\{a, b\}$ is the eligible coupled viability locus. The two branches can supply witnesses by different rules, and their union certifies the corresponding open restriction of the system.

25.3 Context-sensitive contracts

For each p, a proposed supplement may be active only on a specified upward-closed set W. With several such supplements the decision to adjoin a portfolio gives a union of their supply opens. The open of closure over W is full exactly when every $q \in W$ satisfies the supplemented demand table. By Proposition 25.1, this test can be done by listing all failed pointwise comparisons and checking that none lies in W. There is no need to enumerate all open subsets to test a candidate portfolio.

Proposition 25.3 (Sufficient maximal contexts for persistent demand)**.** *Let W be an open subposet of a finite context poset. When each demanded role is demanded everywhere in W, a portfolio supplies all demands on W exactly when it does so at each minimal point of W.*

Proof. Every point in the finite poset W refines at least one minimal point of W. Availability is upward closed for every role. Satisfaction at a minimal point therefore supplies each role at all its refinements. The converse is restriction of full satisfaction. □

For a demand activated only at some contexts, the same argument checks the minimal points of each demand open separately. Indeed satisfaction of that role means containment of its demand open in its supply open, and the generators of an open in a finite poset are its minimal points. This turns the contextual order test into a finite antichain comparison.

25.4 Certificates indexed by context

A demand closure certificate can attach to each minimal demand point a finite production tree that is valid at that point. The tree is then valid at every refinement, since all labels were selected from upward-closed availability. In the local topos reading this

gives a family of witnesses over the principal opens of those minimal points. They cover the demand open, so Theorem 2.2 yields closure there. A viable-state certificate can attach a lasso to each minimal point of its context image. Its finite edges stay active at refinements, so the same finite lasso proves local inhabitation there. Choices at different minimal points need only certify their own regions. If one asks for a single globally coherent witness, that is an additional section problem for the witness family.

25.5 Minimal regions for a supplement

Take a single available supplement on a finite site that supplies every otherwise missing role whenever active. Let F be the subset of contexts with a missing endogenous role. To close globally, its activation open U must contain F. Conversely every open containing F closes the diagram, because the roles are endogenous outside F and the supplement supplies them inside F. The smallest such open is $\uparrow F$. Its minimal points specify a minimal principal-open cover.

With two suppliers each filling part of the deficit, closure imposes a separate cover condition for each role. The context order determines where an activation persists, and the source catalog determines which deficits it covers. In particular, an activation at a minimal missing point covers every refinement there whenever its provider supplies the designated role.

CHAPTER 26

Symmetry and coherent maintenance witnesses

26.1 A groupoid of realizations

A space of witnesses above a role retains automorphisms of its implementations. For a discrete group G acting on a set X, the action groupoid has objects X and arrows $x \to gx$ labeled by $g \in G$. Its classifying space is the homotopy quotient $X//G$. The connected components are the orbits, and the based loop group at x is the stabilizer G_x. The first assertion follows from reachability by arrows of the action. For the second, an automorphism is exactly an element fixing x, with its group composition inherited from G. In particular, the image to a singleton records the existence of a realization, and the fiber retains the orbit and stabilizer data.

Example 26.1 (An exchangeable pair of ports)**.** Suppose a repair uses one of two equivalent ports. The action of C_2 exchanging the ports gives one orbit with trivial stabilizers. The repair using both ports as an unordered pair gives one object with stabilizer C_2. Both profiles can supply the same demand after projecting to a common fact. Their witness fibers are, respectively, contractible and equivalent to BC_2. The distinction comes from computing the automorphism group at a realization.

26.2 Local witnesses on a classifying context

A family over BG given by a finite G-set X has fiber X, and sections are fixed points. Indeed the family is the map $X//G \to BG$. A section picks a fiber point and coherent transport along all loops in BG. Since the fiber is a set, coherence requires exactly that the chosen point be fixed by every group element. A nonempty X yields a cover of the base. Pullback to the canonical point covering BG gives a surjection to that point. Descent then gives the full context image.

Theorem 26.2 (Equivariant kernel and sections)**.** *Let a finite G-set of states carry a G-invariant safe subset and transition relation. The viability kernel over BG is the family associated to the G-stable finite kernel K. Its context image is full exactly when K is nonempty. Its sections are the fixed points K^G.*

Proof. The predecessor of an invariant set is invariant, since translating an edge and its endpoints gives another edge. Thus every deletion stage and the final finite kernel are G-stable. Pullback to the canonical covering point computes the ordinary finite kernel by Theorem 14.4. Equivariance of the stages identifies its descent datum with

the restricted action. The context image and sections then have the descriptions in the preceding paragraph. □

26.3 Equivariant controllers

The existence of an equivariant section of the state family and the existence of an equivariant controller on that family are different fixed-point questions for the group action. For a finite G-set K of viable states, let E_x be the set of admissible labeled outgoing edges from x into K. A controller is a choice $x \mapsto e_x \in E_x$. An equivariant controller requires $e_{gx} = ge_x$.

Proposition 26.3 (Controller descent by stabilizers)**.** *An equivariant controller on a finite viable G-set exists exactly when $E_x^{G_x} \neq \varnothing$ at one representative of every orbit.*

Proof. If the controller is equivariant, for $g \in G_x$ one has $ge_x = e_{gx} = e_x$, so the selected edge is fixed. Conversely choose a fixed edge e_x at each orbit representative and define $e_{gx} = ge_x$. If $gx = hx$, then $h^{-1}g \in G_x$ fixes e_x, so the definition is independent of representative. The resulting rule is equivariant and every selected edge stays in K. □

Take one safe state x and two parallel loop edges interchanged by C_2. The state is fixed, the kernel is inhabited, and every local context admits a control. The edge fiber E_x has no fixed point, so there is no equivariant deterministic controller. A randomized controller assigning probability $1/2$ to each edge is equivariant and safe. The averaging uses a G-invariant probability distribution on the two-element orbit. This example shows how the type of certificate changes when the operational interface asks for a deterministic decision.

26.4 Counting automorphisms in a proof tree

If generators at some profile carry an action of a finite group, tree decorations can have automorphisms from the generator labels and from symmetric permutations of indistinguishable branches. A tree with ordered leaves and no repeated branch profiles has only the automorphisms that fix each branch position. Removing the order and allowing equal subtrees can add permutations. For a corolla with n identical leaves and a fixed root decoration on which permutations act trivially, the unlabeled tree has automorphism group $\mathfrak{S}_n$. Its homotopy orbit gives the component $B\mathfrak{S}_n$. The operation's image still records one available capability. This calculation explains the choice to retain tree spaces before projecting to fact images, even for finite signatures.

CHAPTER 27

Composition of viability regimes

27.1 Independent systems

Let (S_i, V_i, L_i), $i = 1, 2$, be transition graphs over the same context, with synchronized steps that pair arbitrary edges of the two graphs. The product graph has state $S_1 \times_B S_2$, safety $V_1 \times_B V_2$, and edge family $L_1 \times_B L_2$ with paired endpoints. The kernel in this case has a product formula.

Theorem 27.1 (Independent product kernel)**.** *The kernel of the independent product is* $K_1 \times_B K_2$.

Proof. The product of the internal-transition covers over K_1 and K_2 covers $K_1 \times_B K_2$, since products over B preserve effective epimorphisms in each coordinate. Hence the product is postfixed. Conversely project any postfixed certificate for the product to either factor. The internal outgoing edges project to edges with targets in the projected image, and their sources cover the projected image. Each image is postfixed by the simulation comparison in Chapter 16. Hence the projections lie below K_1, K_2, and the certificate itself lies below the product of kernels. □

27.2 An interface of synchronized events

Now attach a label $\lambda_i : L_i \to I$ to each edge and require matching interface labels at a joint step. The allowed edges are $L_1 \times_I L_2$ over B. The safety constraint can additionally be a subobject $W \hookrightarrow V_1 \times_B V_2$ selecting compatible pairs of states. The product kernel is then an upper bound for the kernel of synchronization. This follows from the inclusion into the independent product and Proposition 16.2.

Proposition 27.2 (A sufficient synchronization certificate)**.** *A subobject* $R \hookrightarrow K_1 \times_B K_2$ *contained in* W *lies in the synchronized kernel whenever compatible pairs of edges with endpoints in* R *cover* R *by source. Conversely the synchronized kernel itself has that property.*

Proof. The stated coverage is exactly the postfixed-point property in Theorem 14.1. The converse is the fixed-point equation and the upper bound by the independent kernel. □

This is a criterion for composing a local maintenance controller with a supplier's controller. Each independently viable state must have a compatible next event at the interface. For example, two singleton loops labeled a and b, respectively, each give an inhabited kernel. Over the discrete two-label object, their edge pullback is empty, so the synchronized kernel is empty. Adding a b-labeled loop to the first system gives a

shared edge and restores the product loop. The resulting closure change is due to a synchronization supply, even with identical individual viable-state images.

27.3 An interface relation for storage

Let one system export a token at each step and another consume one. A queue of capacity C is an intermediate state $q \in \{0, \dots, C\}$. A production step increments q unless full, and a consumption step decrements it unless empty. A joint step can be defined with an explicit order of export and consumption. With simultaneous net update, a matched export and consumption leaves q unchanged and is feasible at $q = 0$ if direct transfer is admitted. With consumption-before-export it needs $q \geq 1$, and full-state behavior can differ under export-before-consumption. Each convention is a partial action with a known minimal stock, as in Theorem 9.2. The synchronized kernel thus records an engineering choice about the implementation of the interface.

27.4 Quotients that transport a kernel exactly

For a surjective map $f : S \to S'$ of finite safe graphs, suppose every edge in the source projects to an edge and every edge $f(x) \to y'$ in the target has an edge $x \to y$ with $f(y) = y'$. Safety is pulled back from the target. Then $K = f^{-1}(K')$. The forward inclusion follows by simulation. For the reverse, each state in $f^{-1}(K')$ has a projected successor in K'. The lifting condition gives an edge to $f^{-1}(K')$. Coinduction puts the entire inverse image in K. This gives an exact form of coarse graining for viability, with a statewise lifting certificate.

27.5 Two-sided trace matching

For finite labeled graphs, write $x \; R_0 \; y$ when their state observations agree. Inductively $x \; R_{n+1} \; y$ when their observations agree and every outgoing labeled edge from either has a matching edge from the other with targets related by R_n. This descending chain stabilizes, because there are finitely many pairs. Its stable relation is a bisimulation by construction. Every bisimulation respecting the state observation lies in each stage by induction, so it lies in the stable relation. The finite iteration thus computes the largest two-sided edge-matching relation, giving an exact procedure for behavioral comparison. This is the finite relational form of the transition-logic comparison of Hennessy and Milner. [15]

If safety is among the state observations, kernel membership agrees across this relation. A related viable state has a safe successor in its kernel. A matching edge leads to a related successor. The set of all states related to a viable state is postfixed in the other graph, which proves agreement in that direction. The symmetric matching property proves the reverse. The bisimulation calculation can preserve the viable-state observation even when material, budget, and production data distinguish the records.

CHAPTER 28

Autocatalysis and the restoration of constraints

28.1 A food-generated reaction table

Let $F \subseteq K$ be feedstock in a finite species catalog. A reaction has a finite reactant set, a product set, and a set of species that catalyze it. For a reaction family R, define $G(R)$ as the closure from F under reactants and products, ignoring the catalytic test during that construction. A family R is food-generated when the reactants of each of its reactions lie in $G(R)$. It is reflexively autocatalytic when each reaction has at least one catalyst in $G(R)$. The latter can be a food species or be produced in the family. These are the qualitative conditions of reflexively autocatalytic and food-generated sets in catalytic reaction systems. [24]

Theorem 28.1 (Largest supported autocatalytic family)**.** *Starting from a finite reaction catalog R_0, repeatedly delete any reaction whose reactants are outside $G(R_i)$ in at least one position, or whose catalyst set is disjoint from $G(R_i)$. Simultaneous deletion of all such reactions stabilizes at the greatest food-generated and reflexively autocatalytic subfamily.*

Proof. The map $R \mapsto G(R)$ is monotone by inclusion of rule families. The deletion process is descending and finite. At stabilization, every remaining reaction passes both tests, so the stable subfamily has the stated properties. Any subfamily with those properties contained in R_i passes both tests inside R_i, by monotonicity of G. It survives to R_{i+1}. Induction puts every supported subfamily inside the stable one, proving greatestness. □

If an irreducible family is wanted, inclusion-minimal nonempty subfamilies with the two properties can be found by checking proper deletions. The stable family is a greatest certificate, so it also detects existence of any nonempty supported family. The two fixed points used by the algorithm have different meanings. The inner least fixed point grows the food closure for the current reaction table. The outer greatest fixed point retains the reactions whose production and catalytic conditions survive pruning. Both operators use the same finite catalog.

28.2 Catalyst timing and an initial seed

Take $F = \{f\}$, one reaction $f \to m$ catalyzed by m, and no initial m. Its food closure contains m, so the one-reaction family passes both qualitative tests. In a stock table that requires the catalyst before each use, the transition is $f + m \to 2m$. The initial

stock has zero m, so the transition is disabled. If one m is present, the event deposits a further unit of m and preserves a catalyst for the next event. This compares a support calculation to an operational bootstrap condition with a precise first-event obstruction.

A maintenance model can use the extra m as a replacement, retiring the old one in an installation event. With one feedstock unit supplied per round and a full battery meeting the event guards, the resulting stock returns to its original vector. The proof of indefinite repair is then a repeating stock-and-ledger history. The food-generated calculation, the initial catalyst test, and the renewal cycle now describe three successive certificates for the same network.

28.3 Explicit constraint cycles

A constraint graph on organizational roles has an arrow $x \to y$ when a producer of a replacement for y uses x as a maintained input. Consider a finite catalog with one designated replacement rule for each role and an independent feedstock source for its other inputs. The directed graph records the maintained-input dependencies. Each strongly connected component with more than one vertex, or a self-loop, includes cyclic mutual dependence. The stock semantics adds the quantitative question whether a common round can restore all roles while keeping the catalysts available at every prefix.

Proposition 28.2 (Regeneration with persistent catalysts)**.** *For a finite list of roles $m_1, \dots, m_n$, suppose each replacement event reads a finite set of the current units as catalysts and produces a primed unit m_i'. Each role initially has one current unit and no primed unit. A final event retires all current units and relabels one primed unit of each role as its replacement. With adequate feedstock and an executable ledger for these events, the replacement events can be executed in any order followed by the final event, and the complete round restores all role stocks.*

Proof. Before the final event every current unit persists, because each earlier event has zero net change on current units. Thus every required catalyst is available whenever its event is scheduled. The primed stock acquires one unit in each coordinate. The final event is enabled and replaces the current units with the primed units, restoring the initial vector. This proof applies to every ordering of the replacement events with an executable ledger and the stated feedstock. □

The distinction between preserving a current unit during catalytic use and producing its replacement is the stock expression of closure of constraints. Rosen's closure to efficient causation and the autopoiesis literature motivate an organization that maintains its productive causes. A proof of such maintenance in this table has a constructive sequence of production, installation, and retirement. Both its source requirements and its resulting stock identity are explicit. [21, 22, 19]

CHAPTER 29

Temporal resources beyond periodic forcing

29.1 An arbitrary supply word

Use a given infinite supply and charge word (a_i, c_i) in the arrival-first ledger, and define $d_i = c_i - a_i$. With a full initial battery, the unfolded equation of Chapter 17 implies the exact condition

$$C \;\geq\; \sup_{0\leq h<n} \left(c_h + \sum_{k=h+1}^{n-1} d_k \right).$$

Indeed the initial-prefix term in that unfolding is bounded by the $h = 0$ candidate for every n, and the other terms are exactly the displayed candidates subtracted from C. For the net ledger the exact condition is $C \geq \sup_{0\leq h\leq n} \sum_{k=h}^{n-1} d_k$. These tests use all finite windows. Their finite-horizon versions use the same suprema restricted to $n \leq N$.

Proposition 29.1 (Equal mean and unbounded storage)**.** *There is a bounded, nonnegative arrival word and a constant charge word whose long-run average arrival equals the charge, but whose required capacity is unbounded.*

Proof. Take charge 1 at every step. In block j, take j steps of arrival 2 followed by j steps of arrival 0. Each complete block is balanced. At a point in block j the absolute cumulative surplus is at most j. The number of steps before that block is $j(j-1)$. Hence the average arrival tends to 1. The second half of block j has a deficit window of length j and cumulative deficit j. The net capacity bound grows unboundedly. The arrival-first bound is at least the same value by taking the first charge in that deficit window. □

29.2 Threshold recursion for a finite path

For a finite arrival-first sequence and capacity C, write b_j for the least energy needed at the beginning of step j to finish a prescribed suffix. Set $b_N = 0$. If $C < c_j + b_{j+1}$, then the suffix starting at j is impossible. Otherwise put

$$b_j = \max(0, c_j + b_{j+1} - a_j).$$

If the computed number exceeds C, the suffix is also impossible. The update can reach b_{j+1} or higher exactly when $\min(C, e + a_j) \geq c_j + b_{j+1}$. Solving this inequality proves the recursion by backward induction. For the net ledger the clipping test is $C \geq b_{j+1}$ and the source bound is $\max(0, b_{j+1} + c_j - a_j)$. The different bound on C is the exact effect of the arrival convention.

29.3 Several modes with a common energy coordinate

Consider a finite control graph with arrival-first edge labels a_e, c_e, capacity C, and no charge-dependent guards other than affordability. For a horizon n, let $b_n(x) \in [0, C] \cup \{\infty\}$ be the minimum starting energy permitting n further safe steps from x, with ∞ for an empty upper interval. From each safe x, compute the predecessor of each successor's upper interval by the one-edge formula and take the minimum over edges. From an unsafe state use ∞. This is an exact calculation of the finite-horizon sets by induction, since a finite union of upper intervals is the upper interval above the smallest threshold.

For integer charges and capacities, each $b_n(x)$ is an integer or ∞, and the recurrence stabilizes after at most $|S|(C+1)$ strict stages. Each coordinate is nondecreasing with n, and it can leave at most $C+1$ of the finite values. At stabilization the recurrence yields a greatest fixed point in state space by Theorem 14.1. This threshold implementation stores one number per base state rather than the expanded battery graph. The expanded graph still provides explicit lasso labels and exact viable-state counts.

29.4 Adversarial arrivals

If the controller picks u before a disturbance picks an edge $e \in N(x, u)$, the predecessor of the set of successor thresholds is the intersection of the one-edge source intervals for that control. An intersection of finitely many upper intervals has threshold the maximum of their thresholds. Taking the best control then minimizes over these maxima. This is the same quantifier order as in Chapter 15. The budget dimension turns it into a max–min comparison in the numerical order. A disturbance observed first leads to the opposite nesting, because there is then a separate choice of control for each disturbance.

For example, a single loop may have two energy outcomes under a fixed control with charge 2 and arrivals 3 or 1. The existential battery kernel contains the full state for $C \geq 2$, because the arrival 3 can repeat. For any finite C, the adversary can select arrival 1 at every step, giving cumulative deficit one per step. The robust kernel is empty by the cumulative bound. A second control of charge 1 with deterministic arrival 1 supplies a robust loop for $C \geq 1$. This example connects choice of controller to a robust threshold calculated on the same resource ledger.

29.5 Potentials for budget-independent impossibility

In a finite graph, suppose a function h on vertices and a real $\eta > 0$ satisfy

$$d_e \geq \eta + h(t(e)) - h(s(e))$$

for every safe edge. Then any path of length n has cumulative deficit at least $n\eta + h(x_n) - h(x_0)$, by summing and canceling the internal terms. Since h is bounded on the finite state set, every finite-capacity schedule eventually fails. Such a potential is a short certificate of the cycle condition in Proposition 17.2. Conversely if every simple

cycle in the reachable graph has positive mean deficit, choose η below their smallest mean. Add a virtual source with zero edges to all reachable vertices and assign $h(x)$ the shortest path weight for weights $d_e - \eta$. Every cycle in those weights is nonnegative, so a minimizing path can be taken simple and the minimum is finite. The edge inequality for shortest paths gives the desired potential. If the reachable graph is acyclic, any positive η works by the same construction.

29.6 Exact horizon for a fixed reserve

A one-step deterministic loop of charge c and arrival $a < c$ has deficit $d = c - a > 0$. In a net ledger with starting charge e, each step reduces energy by d, because clipping never applies. It completes exactly $\lfloor e/d \rfloor$ steps. In an arrival-first ledger starting full with $C \geq c$, the first step leaves $C - c$. Every later arrival is below the amount needed to refill the store. Each subsequent step reduces energy by d until it fails. The exact horizon is $1 + \lfloor (C - c)/d \rfloor$. If $C < c$, the first guard fails and the horizon is zero.

At $c = 5, a = 3, C = 10$, these counts are 5 and 3. The two models have the same long-run deficit per completed loop and different horizons at the same capacity. The finite horizons identify the timing convention even when both kernels are empty.

CHAPTER 30

Coarse regimes and testable distinctions

30.1 Three levels of data retention

The full record contains a space of concrete realizations, an attained relation of profile and budget, and a generated subobject of fact availability. There are natural projections from the first to the second and from the second to the third, provided all grades and realizers lie in the budget catalog. A path-level projection additionally sends the transition witness to its state endpoints. The image of each projection is computed from its explicit domain. The resulting lattice of observations can be compared by information retained. When $O_1 = GO_2$, equivalence under O_2 gives equivalence under O_1 by functoriality.

Distinct charges on a unary realization have the same ungraded image and different attained relations. Distinct spaces of realizations with the same possible charges have the same attained relation and different witness fibers. The exchangeable-port example of Chapter 26 gives fibers with different automorphism groups. Distinct route histories with the same state endpoints give one transition relation with different labeled witnesses. These examples prove strictness of each stated information comparison by pairs with the same projected datum and unequal finer invariants.

30.2 Observability of resource thresholds

For two regimes with the same operational labels, define their finite-horizon observation at x to be the set of triples (n, C, a) such that a prescribed arrival word a permits n safe steps at capacity C from a full battery. In a deterministic regime, the exact window formulas compute this observation. For a single step of known arrival a_0, under arrival-first accounting the least full capacity is the step's charge c_0. Thus these one-step full-battery probes identify each immediately accessible deterministic charge. For a path, successive comparisons determine the battery necessary to carry the appropriate deficits across the specified clock.

Proposition 30.1 (Recovering stage charge by reset probes)**.** *Let a finite deterministic phase cycle have one charge per phase and permit resetting the battery to a chosen full capacity at any named phase. The function sending each phase to the smallest capacity admitting one step in the arrival-first ledger equals its phase charge vector.*

Proof. With a full battery of capacity C, nonnegative arrival gives $\min(C, C + a_i) = C$. The guard therefore holds exactly when $C \geq c_i$. Taking its smallest true argument gives

the charge for that phase. □

The test identifies a parameter from a named intervention. When a boundary permits only complete-round observations, multiple phase charge vectors can yield a common total cost or a common kernel context image. The exact threshold tables still separate them when some arrival word has a window that stresses one phase more than another. A concrete separation is the pair of four-step schedules in Chapter 20, whose equal throughputs produce thresholds 3 and 6 under different supply timing.

30.3 Degeneracy of a Boolean classification

In a single-phase full-start regime, suppose an endogenous repair has charge c and arrival a each step. It has structural closure for every pair (a, c). For capacity at least c, viability is inhabited exactly when $a \geq c$. Therefore all $0 < c \leq a$ share the same two affirmative Boolean judgments for sufficiently large capacity. Within that region, different c's give distinct budget relations and different minimal full capacities. The continuous family of threshold values gives many structures in one Boolean class. This illustrates the usefulness of the richer invariant $\mathcal{J}$.

30.4 A comparison of research programs

Autopoiesis makes a network of component production and boundary production central to the individuation of a living system. Closure of constraints distinguishes the persistence of constraints at a relevant time scale from the processes they channel. The formal roles in this manuscript give those distinctions an implementation as catalyst guards, production labels, and replacement obligations. The two fixed points treat the ability to produce a replacement and the ability to repeat maintenance as mathematically different operations. [18, 19, 20]

The open-process literature gives categories whose composition connects exposed interfaces. The operadic coupling here uses an explicit common interface and a realization image, so a mixed tree supplies a new witness with specified internal sources. Reaction-network semantics adds dynamics to a network and supports compositional comparison of open systems. This paper keeps a reaction or process realization available at the level of the proof certificate before a finite state compilation. [9, 10]

Viability theory computes controlled invariance inside a region and supports both continuous and discrete regimes. The finite lasso, disturbance game, and exact battery calculations implement the state component of the organization semantics. In a categorical thermodynamic account, viability localization selects a chosen observational resolution. Here the least and greatest fixed points are computed on the explicit catalogs and state graphs, and an observation functor can then be applied to those calculations. [16, 17, 31]

The distinction between a maintained cell, a host-dependent replicator, and an

industrially sustained device is specified by the demand list, the supply of realizers at the boundary, and the ledgered transition graph. A comparison of the corresponding invariants consists of checking their typed witnesses, and exact model calculations test the comparisons across multiple boundaries and operating regimes.

30.5 Choosing the resolution of comparison

At a fixed boundary, a measured change of substrate changes $\mathrm{Supp}(X)$. A change in the supply of machinery changes a production or provenance datum. A change in exergy rhythm can change the viable kernel with the same productive catalog. These changes can be tested separately. A comparison across boundaries uses the selected interface map and the corresponding coupled interpretation, so a newly introduced proof vertex can be assigned to its source rather than attributed to the original system.

For a design stated in terms of independent repairs, the natural comparison reports demanded roles, their attained budgets, and the classes of starts that repeat the associated rounds. When the design also specifies common resources or synchronized suppliers, the same report is computed in the combined state graph. The product and synchronization calculations give the exact test for that change in specification. The following chapters extend the construction to live cohorts, adverse timing, and certificates of bounded computational size.

CHAPTER 31

Production conditioned by a viable cohort

31.1 A registry of available capabilities

A cohort of mutually supporting subsystems can share a catalog of repair operations. Membership in the cohort determines which subsystems supply operations to the catalog. Continued membership in turn requires that each subsystem have a feasible maintenance transition using the shared catalog. This gives a nested fixed point in which the production presentation is recomputed as the candidate cohort changes.

Use a finite constant state catalog S over a context and a finite fact catalog K. Each seed and primitive rule has either no owner or one owner in S. The owned seed or rule is admitted for a candidate subobject $A \hookrightarrow S$ at the contexts where its owner lies in A. Let $G(A)$ be the generated production image of the resulting presentation on K. Each edge e of a safe transition graph has a finite prerequisite list U_e in K. The edge is enabled for the cohort A when every listed fact is in $G(A)$. The predecessor $P(A)$ uses those enabled edges and requires their targets to belong to A. Thus

$$P(A) = V \ \wedge \ \mathrm{im}\Big(\{e \mid t(e) \in A,\ U_e \subseteq G(A)\} \xrightarrow{s} S\Big)\,.$$

The stock-and-budget guards can be part of the original edge catalog. The additional test on U_e checks the capabilities of the candidate cohort.

31.2 Joint maximality

Call a pair (A, C) certified when $C \leq G(A)$, $A \leq V$, and every state of A locally has an edge into A with all its prerequisites in C. The first condition witnesses the organizational supply of the pair. The last witnesses its viable continuation.

Theorem 31.1 (Greatest certified cohort)**.** *The pair $(A_*, G(A_*))$, with $A_* = \nu P$, is certified and contains both coordinates of every certified pair. In finite sets it is computed by descending deletion of candidate states, recomputing the least production fixed point at every step.*

Proof. Enlarging A enlarges the admitted seed and rule families, so it enlarges $G(A)$. It also enlarges the allowed targets. The image in P is therefore monotone. The greatest fixed point exists by the coinduction argument of Theorem 14.1. Its fixed-point equation gives the required local outgoing cover using $G(A_*)$, so the pair is certified. For any certified (A, C), the prerequisite relation $C \leq G(A)$ puts each of its witnessing edges in

the formula for $P(A)$. Hence $A \leq P(A)$, and coinduction yields $A \leq A_*$. Monotonicity gives $C \leq G(A) \leq G(A_*)$. In a finite set the descending chain from V stabilizes, and its stabilized value is the greatest fixed point. Each evaluation is a least-fixed-point computation on the indicated seed and rule catalog. □

The common catalog on the greatest pair consists of finite derivations rooted in the admitted owner set. An edge in that pair uses a finite tree for each prerequisite. A local proof of membership therefore consists of a coinductive state certificate and an inductive family of repair witnesses relative to it. The two parts refer to the same owner set.

31.3 Uniform finite transport

When S is constant with N elements, each stage $P^j(V)$ commutes with geometric inverse image. To see this, pull back the owner map and membership test into A, then apply Corollary 5.4 to the admitted rule families. The prerequisites on an edge have finite length, so the edge test and predecessor use finite products and images. For a varying length, the test is the union of those operations over the finite arity classes in the catalog.

The chain stabilizes by stage N in every topos for this constant-state presentation. Group rules by owner, output, and set of premises, and group edges by endpoints and set of prerequisite facts. Each type contributes its image in B. Thus the calculation uses a finite table of subobjects of B, even when one type has an infinite family of realizers. All expressions in the stage calculation are built from finite conjunctions and disjunctions of these flags. The distributive lattice of positive expressions in finitely many variables is tested by its two-valued assignments. In detail, write a formula as a disjunction of conjunctions of subsets of the variables and remove any subset containing another displayed subset. On an assignment with precisely one of the minimal subsets true, that formula is true. Thus two formulas with the same truth table have the same minimal subsets and give the same value in every distributive lattice. The stage-N and stage-$N+1$ formulas agree on every two-valued assignment, by the finite descending-chain bound. They consequently agree as subobjects. The stable kernel and its image over the context commute with inverse image by the finite formula.

31.4 A cascading loss of supply

Take states x, y, z and facts a, b. The seed a is owned by y, the seed b by z, and there are no other productions. Give x a loop requiring a, y a loop requiring b, and z no outgoing edge. Starting with $\{x, y, z\}$, the catalog supplies a, b and the first predecessor yields $\{x, y\}$. At the next stage b is unavailable, so the predecessor yields $\{x\}$. Now a is unavailable and the third stage is empty. Holding the initial supply fixed would have preserved both loops. The successive emptying of the catalog proves loss of mutual support in the cohort model.

Adjoin a host-owned rule with no cohort owner producing b. The first stage still

deletes z. Now y's loop has its required b, and x's loop uses a from y. The remaining two states form a certified pair. This host supplement restores the joint fixed point by adding a production with a different ownership type.

31.5 Removal certificates in an owned catalog

A state can lose its last successor inside the candidate cohort, or its remaining successors can require a fact outside the recomputed production image. In the second case a finite closed set omitting the fact certifies the shortage by Chapter 6. Each lost source owner already has an earlier removal certificate. The final certificate can thus link state deletions to lost transitions and missing outputs.

In the three-state example, z is removed at the first stage by its empty successor list. The missing set $\{b\}$ then blocks the loop at y. The missing set $\{a, b\}$ blocks the loop at x at the following stage. The order of this certificate is independent of how the finite proof trees for still-available facts were stored.

CHAPTER 32

Worst-case timing on a disturbance graph

32.1 A supply graph with arbitrary continuation

Let a finite directed graph describe the operating phases and the allowed disturbances. Every reachable vertex has at least one outgoing edge. Each edge e specifies nonnegative arrival a_e and charge c_e, with deficit $d_e = c_e - a_e$. The environment can follow any infinite path from a fixed initial vertex x. A full battery of capacity C is robustly viable when the guards of the arrival-first ledger hold on every such path.

Let R be the graph reachable from x. For $y \in R$ put

$$L(y) = \sup \left\{ \sum_{e \in w} d_e \;\middle|\; w \text{ a finite path from } y \right\}.$$

The empty path contributes 0. This is the maximum possible accumulated deficit after the first edge of a capacity-critical window.

Theorem 32.1 (Robust capacity for arbitrary timing)**.** *If R contains a cycle of positive total deficit, every finite capacity fails on some disturbance path. Otherwise $L(y)$ is the maximum over simple paths from y, and the least full-start arrival-first capacity robust against every disturbance path is*

$$C_{\mathrm{rob}} = \max_{e \in R} \big(c_e + L(t(e)) \big).$$

In the net ledger the corresponding capacity is $\max_{y \in R} L(y)$.

Proof. A positive-deficit cycle can be reached and then repeated. The total required deficit on its repetitions grows without bound, which is impossible from finite initial energy by the cumulative balance. Otherwise, delete a closed segment whenever a path repeats a vertex. Its deficit is nonpositive, so deletion weakly increases the path's total deficit while leaving its starting vertex unchanged. A simple path therefore gives a sum at least that of every path, proving the description of $L(y)$.

For a fixed infinite path the unfolding in Chapter 29 makes each arrival-first guard equivalent to the collection of window inequalities beginning at one of its visited edges. Every such window has requirement $c_e + \sum_{f \in w} d_f$, with w a path after e. These are bounded by C_{rob}, which proves sufficiency. For necessity, choose a reachable edge e and a path from its target attaining $L(t(e))$. A path from x to the source, followed by the edge and that maximizing path, extends to an infinite path since the graph has no reachable

dead ends. Its specified window imposes the threshold $c_e + L(t(e))$. The net ledger uses deficit sums on all windows, including the empty one. A path from a reachable vertex attaining its L supplies the necessary window, and the same upper bound gives sufficiency. □

32.2 A finite critical-window certificate

With n reachable vertices and no positive-deficit cycle, a path attaining $L(y)$ has at most $n - 1$ edges. Starting with $L_0(y) = 0$, compute

$$L_{j+1}(y) = \max\Big(0, \max_{e:s(e)=y} (d_e + L_j(t(e)))\Big).$$

The right side separates the empty path from paths with a chosen first edge. Induction identifies L_j with the maximum over paths of length at most j. The simple-path bound gives $L_{n-1} = L$. Recording the selected edge at each maximum gives an exact certificate for the critical window. The arithmetic and critical path are finite even though the environment is allowed an unbounded number of different infinite schedules.

Example 32.2 (Correlated low supply)**.** Let $x \to y$ have arrival 0 and charge 3, and let $y \to x$ have arrival 5 and charge 1. Add a loop at x with arrival 2 and charge 2, and a loop at y with arrival 0 and charge 0. Both loops have deficit zero, and the two-cycle has deficit -1. The largest deficit from x is 3 and from y is 0. The four arrival-first candidates are $3, 4, 5, 0$, respectively for the two-cycle edges and the two loops in that order. Hence the robust arrival-first capacity is 5, while the robust net capacity is 3. A critical window consists of the charge-two loop at x followed by the zero-arrival edge to y. A rule that repeatedly treats the two low-supply edges as unconstrained alternatives would permit different timing words. The graph calculation keeps their phase constraints.

32.3 Comparison with selectable operation

For the same graph, a controller choosing a favorable successor needs some reachable cycle with nonpositive deficit, by Proposition 17.2. Robust operation against every disturbance requires every reachable cycle to have nonpositive deficit and every reachable vertex to have a continuation. The quantifiers produce different conditions even at unbounded design capacity. A single positive-deficit cycle reachable in the presence of a second balanced cycle gives a viable selectable regime and an empty robust full-start capacity set. The two properties compare the same physical charges and distinct information about who selects the next transition.

CHAPTER 33

Decision procedures and witness size

33.1 Linear closure on an explicit rule table

An explicit finite rule table lists every premise occurrence and every output. To compute the unweighted closure, maintain a counter of unfulfilled premise occurrences for each rule and an incidence list of the occurrences using each fact. Initialize a queue with the seeds and outputs of nullary rules. Whenever a fact is first reached, process its incidence list and decrement the corresponding counters. A counter reaching zero enqueues the rule's output if that output is still unreached.

Proposition 33.1 (Closure work bound)**.** *The queue procedure computes the generated fact set in time linear in the number of facts, rules, and premise occurrences.*

Proof. Every counter is decremented once for each processed occurrence, and a fact is processed only the first time it is reached. Every enqueued conclusion has all its premises already reached, so it belongs to the generated set. When the queue empties, all seeds have been reached and every rule with all premises reached has had its counter reach zero. The reached set is closed. The two inclusions give equality with the least fixed point. Initializing counters and adjacency lists uses linear time, and each entry is subsequently processed a bounded number of times. □

The first producing rule for a reached fact can be stored as a directed acyclic certificate, with edges to earlier reached premises. Reading a certificate as a tree may duplicate those earlier pieces. For the single seed a_0 and rules $(a_i, a_i) \to a_{i+1}$, a proof of a_n expands to 2^n leaves, by induction on n. The shared certificate stores n rule applications and their repeated links. The expanded grade, with a unit charge at each leaf and zero charges at rules, is exactly 2^n, computed by doubling along the shared certificate. In this model storage of a proof reference and the physical accounting for its two branches are separate operations.

33.2 Linear deletion on an expanded graph

For the finite existential kernel, start from the safe graph, store safe outdegrees, and queue vertices of outdegree zero. Deleting such a vertex removes all incoming edges from surviving vertices, so update their outdegrees and queue newly dead vertices. When the queue empties, every remaining vertex has a surviving successor. It is postfixed, and every earlier deleted vertex has a rank of failure given by one plus the maximum rank of its deleted safe successors, with empty maximum zero. Induction on the deletion order

proves these certificates. Each vertex and edge is processed a bounded number of times, giving linear work in the explicit graph.

For an integer battery of capacity C, the lifted graph has at most $(C + 1)|S|$ states and $(C + 1)|L|$ edges for deterministic updates given an edge label. Thus the deletion algorithm has linear work in this expanded representation. When capacities are written in binary the bound is polynomial in their numerical value. The threshold recurrence of Chapter 29 stores fewer coordinates and supplies the same minimum start-energy calculation. A separate finite lasso can be verified by checking its labeled resource update at each step.

33.3 The cost of selecting external sources

Theorem 33.2 (Portfolio selection)**.** *The decision problem of whether at most k supplements suffice for all demands in a finite Horn presentation is NP-complete, even when every supplement supplies only facts and there are no nonnullary rules.*

Proof. A list of chosen supplements is a certificate. Its length and the generated closure are checked in polynomial time by the queue procedure. For hardness, reduce Set Cover to the selection problem. Take one fact for each universe element, demand all facts, and for each available subset introduce a supplement with a nullary producing rule for each of its elements. The endogenous catalog supplies none of the demanded facts. For a chosen portfolio the generated set is exactly the union of those subsets. A portfolio of size at most k therefore supplies the demands exactly when the original instance has a cover of size at most k. The construction is polynomial and Set Cover is NP-complete. [32] □

The corresponding search can be verified using the minimal portfolio computation of Chapter 24. There can be exponentially many inclusion-minimal portfolios. Take n demanded facts and two distinct named sources for each, each source supplying only its associated fact. Every minimal portfolio chooses exactly one from each pair, giving 2^n minimal portfolios. The model still admits a direct fact-by-fact proof of each portfolio. This separates the cost of enumerating all minimal designs from the cost of checking one.

33.4 Certificates at a boundary

A shared proof is checked by testing every premise reference and its rule profile. A missing set is checked by testing the premise intersection for each producing rule. A viable lasso is checked at every edge guard and at the closing state. Deletion ranks are checked by comparing every removed state’s rank to those of its safe successors. These tests verify the finite record at the chosen boundary.

CHAPTER 34

Common certificates for local organization

34.1 One witness for a finite conjunction

Let B be a context in a higher topos. Fix a finite list of constant demand labels $r_1, \dots, r_k$. For each i let $W_i \to B$ be the finite-tree witness object for r_i, pulled back along its label from a budget-graded production presentation. Let $Q \to B$ be the eligible part of the kernel of a finite round graph, with state cardinality bounded by N. In this section the round graph already contains the admitted production, stock, and budget checks on its edge labels. Let $\mathcal{P}_N \to B$ be the object of safe paths of length N starting in the eligible region. The endpoint map to their starting states covers Q by the horizon theorem and the uniform finite kernel formula.

Theorem 34.1 (Common witness object)**.** *The image over B of*

$$\mathcal{C} = W_1 \times_B \cdots \times_B W_k \times_B \mathcal{P}_N$$

is the context locus of simultaneous production feasibility for the listed demands and existence of an eligible viable start. The object $\mathcal{C}$ and its image commute with geometric inverse image transporting the presentation and finite state bound.

Proof. Write $L_i = \mathrm{im}(W_i \to B)$ and $L_V = \mathrm{im}(\mathcal{P}_N \to B)$. The product of the effective epimorphisms onto these images covers their fiber product in the slice, and the latter product is their meet as subobjects of B. Thus the image of $\mathcal{C}$ is $\bigwedge_i L_i \wedge L_V$. Each L_i is the production-feasibility locus by Theorem 5.3 and the attainable-budget construction. The path image is the eligible viability locus by Lemma 14.2 and Theorem 14.4. This proves the first assertion. Inverse image preserves the witness objects by Corollary 5.4 and the budget relation, and preserves the finite path object because it is a finite limit of the state and edge diagrams. It preserves their product and the image. □

The product gives a single cover on which all listed witnesses and an eligible path are available together. The path itself is already labeled by the operation witnesses actually executed on that path. The independent components W_i give feasibility of the designated roles at the chosen static budgets. The distinction between the two occurrences of production data makes it possible to add one budget for a static audit and a separately updated ledger for the actual rounds.

34.2 Several recurring obligations on a finite graph

Suppose a finite safe graph has finitely many kinds of marked completion. A path satisfies the recurrence condition when it traverses an edge of each kind infinitely often.

Proposition 34.2 (Recurring completion certificate)**.** *A state satisfies the recurrence condition along some infinite safe path exactly when it reaches a strongly connected component containing an internal edge of every specified kind and containing a nonempty closed walk. A finite prefix followed by one repeated closed walk supplies a witness.*

Proof. For necessity, the vertices visited infinitely often form a strongly connected set as in Chapter 16. For each kind, at least one of its finitely many edges occurs infinitely often, so that edge has both endpoints in this set. An infinite path in the finite set repeats a vertex and contains a closed walk. For sufficiency, choose an internal edge of each kind. In the component, join the target of each chosen edge to the source of the next one and join the last to the first. The resulting nonempty closed walk includes every chosen edge. With an empty list of kinds use the stipulated closed walk. Reach the walk from the start and repeat it indefinitely. □

This proposition applies to a lifted stock-and-energy graph as well as to a graph of base modes. The finite prefix and closed walk then agree in every stock and phase coordinate at their interfaces. If production labels are retained on the chosen edges, concatenating them records the full ledgered witness of recurring completion.

34.3 Separating an aggregate from its projection

A record of the kinds of completion that appear somewhere in a safe graph is a union over its edges. Proposition 34.2 tests the occurrence of all kinds within one reachable recurrent component. For example, give two disconnected safe loops different completion kinds, and make both initial. The aggregate union contains both kinds. Each infinite history stays in one loop and supplies exactly one kind repeatedly. Adding one transition in each direction joins the loops into a strongly connected component and yields a walk supplying both kinds. The new transitions change the recurrence relation while leaving the aggregate list of available kinds unchanged.

This comparison gives one more use of boundary information. A proof of recurrence lists the interface events that connect completions, and a proof of its failure can list the reachable strongly connected components and the absent mark in each. When the repertoire of executable states is finite over a context, the comparison can also be made using finite paths and loops as local witnesses. Its source image gives the exact region on which the joint completion schedule exists.

CHAPTER 35

Finite verification and reproduction

35.1 Encoding and comparison

The finite atlas gives rule catalogs, graph edges, and boundary schedules in the same notation used in the proofs. All counts, budget thresholds, and viable-state computations use integer arithmetic. Probability tables use rational arithmetic. Each periodic case uses both ledger conventions and checks the full threshold against direct iteration. Each portfolio case records its explicit seeds, primitive rules, costs, and source labels. Each graph case displays the original transitions and computes the derived graph using the given stock, guards, and arrival convention.

The numerical calculations are instances of the mathematical models. Their inputs specify maintenance roles and interfaces through short event labels, and the tables can be regenerated from the displayed data. Algorithms are given below. Roman letters for states or facts are local to a case. A state named x in one case is independent of a state with the same name in another.

35.2 Inductive certificates

Algorithm 35.1 uses simultaneous rounds of rule firing. This convention fixes the stage numbers in the atlas. A program can store a witness for each new fact by recording a rule and references to the immediately preceding stages of its prerequisites. Reading those references as a finite tree yields the proof of Theorem 5.3. A fact seeded at stage zero uses its seed label. For a budgeted comparison use the recurrence of Theorem 11.1 and store the minimizing rule and child references at each height.

Algorithm 35.1 (Finite closure with ranks)**.** From seed set S, rule table E, and fact set K, compute the least stage of each fact.

```
active = set(seeds)
rank = {k: 0 for k in active}
stage = 0
while True:
    new = {out for ins, out in rules
           if all(k in active for k in ins)} - active
    if not new:
        break
    stage += 1
    for k in new:
        rank[k] = stage
```

```
    active |= new
```

In a portfolio table, each retained grade comes with a provenance set and a certificate. The recurrence starts with the seed entries. Its step combines each root's charge with all tuples of entries at the input labels. Sources are combined by set union. A budget query for a fact asks whether one retained entry has both sources and costs below the query. Budget vectors and source sets are compared componentwise in their respective orders. For each finite catalog, the simultaneous recurrence runs through height $|K|$, with the bound justified by Theorem 11.3. Entries with the same sources and grade can be stored once for the decision procedure.

35.3 Coinductive certificates

For a finite graph, the following descending recursion records both the kernel and the horizon at which a safe state first disappears. The initial safe set V is stage zero. A state removed in the first deletion has zero safe successors. A state with deletion rank j admits $j - 1$ further safe transitions and fails to admit j, by Lemma 14.2. The corresponding ranks for the support game measure the horizon of a forced exit.

Algorithm 35.2 (Kernel and deletion ranks)**.** Given safe states and a successor table, compute the kernel.

```
active = set(safe)
removed = {}
stage = 0
while True:
    keep = {x for x in active
            if any(y in active for y in successors[x])}
    if keep == active:
        break
    stage += 1
    for x in active - keep:
        removed[x] = stage
    active = keep
```

The loop is equivalent to using V in each predecessor stage. Monotonicity and descending induction imply that every state admitted by the next application already belongs to the current A. For the robust game, replace the condition on successors by existence of a control with a nonempty support all of whose members remain active. For probability tables use the finite-horizon recursion of Chapter 15, initializing by the safety indicator and computing each next value over the original control table. The game and the probability calculation then give independent checks of all probability-one entries.

A lasso certificate can be found by choosing at each retained state an edge staying in the retained set and following the choices from the designated start. The first repeated vertex splits the trace into the prefix and a nonempty period. Each edge of that lasso can be verified using the original table. For the expanded battery graph, a vertex consists of the base state and the current stock. The label of a transition specifies arrival and charge, which determine the next stock.

35.4 A direct ledger verification

The next procedure works for a single fixed periodic word with integer entries. It tests a proposed capacity and starting charge by simulating until a guard fails or a state repeats. A repeated state includes the phase and charge. The deterministic update after a repeat is the same as it was on the first visit. If no guard fails, there are at most $m(C+1)$ distinct states, so one of them repeats. This is an independent finite calculation of the viability values predicted by Theorem 17.1.

Algorithm 35.3 (Periodic full-start feasibility)**.** Use a, c for the periodic arrivals and charges. Run the update with either clipping order.

```
phase, energy = start_phase, start_energy
seen = set()
while (phase, energy) not in seen:
    seen.add((phase, energy))
    if arrival_first:
        energy = min(capacity, energy + a[phase]) - c[phase]
    else:
        energy = min(capacity, energy + a[phase] - c[phase])
    if energy < 0:
        return False
    phase = (phase + 1) % len(c)
return True
```

From a full battery each feasible deterministic period enters a periodic energy trace. The transient and period printed in the atlas are obtained by retaining the index of the first occurrence of each state. A failure entry reports the number of completed steps before the first unsatisfied guard. This convention agrees with the finite viability horizons.

35.5 Checking the atlas by independent identities

The following comparisons give internal checks on the exact data.

1. In each periodic case, compare the circular-window threshold and the phase-prefix formula with the complete finite graph for capacities near the threshold. For net positive deficit the cumulative inequality gives an independent obstruction.

2. In each portfolio case, enumerate source subsets, run the exact closure recurrence, and compare the successful subsets with supersets of minimal proof portfolios. In the weighted calculation, every retained grade is verified as the sum of its labeled rule and seed grades.
3. In each control case, compare the probability-one horizon values with the support-game deletion stages. Check the existential ranks by direct graph deletion.
4. In each selectable ledger case, check both the expanded-state deletion algorithm and the threshold recurrence that stores one minimum per base state. The two computations use different representations of the viability subobject.
5. In each stock case, compare minimal-stock and displacement formulas with direct event simulation on the proposed minimal stock, then lower each positive initial coordinate by one and locate a failed guard.
6. In each contextual case, compute the closure open by deleting predecessors of pointwise failures. Independently check it as the largest open contained in the pointwise success set.

These checks are exact computations on the displayed finite models. The mathematical proofs establish the formulas before any particular case table is evaluated.

CHAPTER 36

A guide to the finite atlas

36.1 How the model sheets are organized

Each case specifies a complete small presentation and separates its input data from the corresponding certificates. Cases are grouped by which coordinate of the bounded-system record varies. The schedules of periodic supply use the same types of operation under different exergy rhythms. The portfolios compare whole source sets and exact graded witnesses. The game tables compare possible and robust continuation on a common transition model. The selectable ledgers combine mode choice with the exact charge update. The stock cases retain token multiplicities and explicit retirement. The contextual cases evaluate local closure and finite viability on a finite frame.

36.2 Conventions for compact tables

A dagger in a graph denotes a failure state outside the safe set. An infinity sign in a budget or kernel table denotes an empty admissible set for that test. When a table uses K_i for a safe graph's stages, the stage index counts transitions already allowed, with K_0 all safe states. In a periodic table, (i, e) means phase i at stored charge e, before the arrival for that phase. The notation C denotes capacity, M the net-ledger threshold, and M_{a} the arrival-first threshold. A column of prefix thresholds gives the minimum starting charge at each phase at any sufficiently large capacity.

For portfolio rules, a dash in the source column indicates endogenous production. Each other source label activates every rule marked with that source when the corresponding supplement is adjoined. Empty premises are printed as 1, for a nullary production. The target r records a completed obligation. Two entries with the same result can be joined by a later rule only when the premise list names that result. For stocks, a vector has the order of species given on the sheet. Initial and final stocks in a repeat test use that same order.

For contextual models, the arrows between contexts generate the order by reflexivity and transitivity. Demands and supplies are principal opens or unions of principal opens. The logical closure locus is an open even when the current pointwise successes fail to form an open. Distinct loops and transition paths can supply local evidence in distinct contexts. The tabulation uses the implication formula of Chapter 25 and the geometric finite kernel formula of Chapter 14.

36.3 The worked example as a certificate

A comparison sheet should be read as a short proof with a stated finite input. Its arithmetic steps follow the displayed rule or edge table. The witnesses of success are proof trees, prefixes, cycles, or minimal opening stocks. The obstructions are missing prerequisite labels, strict capacity inequalities, support-game deletion ranks, or downward-closed sets of failing contexts. These certificates compare regimes on one catalog. They can also be used separately as examples of the inductive and coinductive constructions of the manuscript.

36.4 Comparing adjacent sheets

An adjacent pair of periodic sheets can hold the charges fixed while shifting supply. The operation catalog is identical and each changed threshold is attributable to a deficit window. The source sheets instead alter the operations supplied and keep the two charge coordinates fixed. They identify a supplier by membership in a minimal portfolio or in a deletion set. In the stock family the distinction between before and after delivery is encoded as a new event order, and the computed initial stock records which order draws on a preexisting component.

A comparison across families uses the same types of evidence at different resolutions. A typed host rule can supply a missing intermediate on a portfolio sheet. The induced repair edge can then be entered into a selectable ledger and tested at a particular capacity. If that edge is active on part of the context site, its active open is an input to the contextual kernel. This sequence moves from a production witness to a viable trajectory and then to its region of validity.

CHAPTER 37

Periodic maintenance schedules

The first family varies charge timing and environmental supply in a fixed repeated schedule. Each step has an arrival, a repair charge, and a phase update. The only state variables on these sheets are phase and stored energy. The finite graph at capacity C therefore has $m(C+1)$ possible states.

The last two rows of each table derive the two capacity calculations of Theorem 17.1. The largest prefix deficit is a starting-energy requirement. The arrival-first candidate includes the charge at the beginning of a window, followed by its largest later deficit. The first capacity is the maximum of the prefix row, provided total deficit is nonpositive. The second is the maximum of the candidate row under the same throughput test.

The repeating energy traces certify exact admissible cycles. Each trace is computed by executing the guarded update from a full start until the first repeated phase-charge pair. The phase always advances by one, so the energy trace, together with its start at phase zero and period split, specifies its entire path.

MODEL A.01

Evenly distributed repair

The cellular repair order receives a small arrival at each phase. The high-cost fabrication step is followed by installation.

Phase ledger

Phase	0	1	2	3
Charge	1	2	3	1
Arrival	2	2	2	1
Deficit	-1	0	1	0
Largest prefix deficit	0	1	1	0
Arrival-first candidate	2	3	3	1

Capacity certificate

The totals are charge 7 and arrival 7, giving $D = 0$. The full-start capacities are $M = 1$ for the net ledger and $M_{\mathrm{a}} = 3$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $3 \to 2 \mid 1 \to 0 \to 0 \to 1 \to 1$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 2, a full phase-zero start completes 2 safe transitions before the guard fails at phase 2.

Viability profile

At capacity 3, the viable starting charges by phase are $[0, 3]$, $[1, 3]$, $[1, 3]$, $[0, 3]$.

The kernel contains 14 phase-charge states at that capacity, and 18 when capacity increases by one.

Capacity sensitivity

Capacity	1	2	3	4	5
Net-ledger viable states	6	10	14	18	22
Arrival-first viable states	0	0	14	18	22
Total phase-charge states	8	12	16	20	24

A critical arrival-first window begins at phase 1. Its leading charge is 2 and the following 1 phase contributes net deficit 1. These terms give the exact threshold in the capacity certificate.

MODEL A.02

Two clustered arrivals

The same four cellular charges are paid from two pulses. The middle phases draw on stored supply.

Phase ledger

Phase	0	1	2	3
Charge	1	2	3	1
Arrival	4	0	0	3
Deficit	-3	2	3	-2
Largest prefix deficit	2	5	3	0
Arrival-first candidate	6	5	3	3

Capacity certificate

The totals are charge 7 and arrival 7, giving $D = 0$. The full-start capacities are $M = 5$ for the net ledger and $M_{\mathrm{a}} = 6$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $6 \mid 5 \to 3 \to 0 \to 2 \to 5$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 5, a full phase-zero start completes 2 safe transitions before the guard fails at phase 2.

Viability profile

At capacity 6, the viable starting charges by phase are $[2, 6]$, $[5, 6]$, $[3, 6]$, $[0, 6]$.

The kernel contains 18 phase-charge states at that capacity, and 22 when capacity increases by one.

Capacity sensitivity

Capacity	4	5	6	7	8
Net-ledger viable states	0	14	18	22	26
Arrival-first viable states	0	0	18	22	26
Total phase-charge states	20	24	28	32	36

A critical arrival-first window begins at phase 0. Its leading charge is 1 and the following 2 phases contribute net deficit 5. These terms give the exact threshold in the capacity certificate.

MODEL A.03

A pulse before all events

A single early arrival supplies all four cellular events. Any capacity below the threshold spills part of the energy needed later.

Phase ledger

Phase	0	1	2	3
Charge	1	2	3	1
Arrival	7	0	0	0
Deficit	-6	2	3	1
Largest prefix deficit	0	6	4	1
Arrival-first candidate	7	6	4	1

Capacity certificate

The totals are charge 7 and arrival 7, giving $D = 0$. The full-start capacities are $M = 6$ for the net ledger and $M_{\mathrm{a}} = 7$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $7 \mid 6 \to 4 \to 1 \to 0 \to 6$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 6, a full phase-zero start completes 3 safe transitions before the guard fails at phase 3.

Viability profile

At capacity 7, the viable starting charges by phase are $[0, 7]$, $[6, 7]$, $[4, 7]$, $[1, 7]$.

The kernel contains 21 phase-charge states at that capacity, and 25 when capacity increases by one.

Capacity sensitivity

Capacity	5	6	7	8	9
Net-ledger viable states	0	17	21	25	29
Arrival-first viable states	0	0	21	25	29
Total phase-charge states	24	28	32	36	40

A critical arrival-first window begins at phase 0. Its leading charge is 1 and the following 3 phases contribute net deficit 6. These terms give the exact threshold in the capacity certificate.

MODEL A.04

A pulse after conversion

The cellular period moves the pulse from transport to fabrication. This shifts the viable starting charges between phases.

Phase ledger

Phase	0	1	2	3
Charge	1	2	3	1
Arrival	0	0	7	0
Deficit	1	2	-4	1
Largest prefix deficit	3	2	0	4
Arrival-first candidate	3	2	7	4

Capacity certificate

The totals are charge 7 and arrival 7, giving $D = 0$. The full-start capacities are $M = 4$ for the net ledger and $M_{\mathrm{a}} = 7$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $7 \to 6 \to 4 \mid 4 \to 3 \to 2 \to 0 \to 4$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 6, a full phase-zero start completes 5 safe transitions before the guard fails at phase 1.

Viability profile

At capacity 7, the viable starting charges by phase are $[3, 7]$, $[2, 7]$, $[0, 7]$, $[4, 7]$.

The kernel contains 23 phase-charge states at that capacity, and 27 when capacity increases by one.

Capacity sensitivity

Capacity	5	6	7	8	9
Net-ledger viable states	15	19	23	27	31
Arrival-first viable states	0	0	23	27	31
Total phase-charge states	24	28	32	36	40

A critical arrival-first window begins at phase 2. Its leading charge is 3 and the following 3 phases contribute net deficit 4. These terms give the exact threshold in the capacity certificate.

MODEL A.05

Host preparation and export

A host operation prepares the machinery, followed by a viral preparation and assembly. Supply is concentrated in the latter two phases.

Phase ledger

Phase	0	1	2
Charge	1	2	3
Arrival	0	4	2
Deficit	1	-2	1
Largest prefix deficit	1	0	2
Arrival-first candidate	1	4	4

Capacity certificate

The totals are charge 6 and arrival 6, giving $D = 0$. The full-start capacities are $M = 2$ for the net ledger and $M_{\mathrm{a}} = 4$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $4 \to 3 \mid 2 \to 1 \to 0 \to 2$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 3, a full phase-zero start completes 3 safe transitions before the guard fails at phase 0.

Viability profile

At capacity 4, the viable starting charges by phase are $[1, 4]$, $[0, 4]$, $[2, 4]$.

The kernel contains 12 phase-charge states at that capacity, and 15 when capacity increases by one.

Capacity sensitivity

Capacity	2	3	4	5	6
Net-ledger viable states	6	9	12	15	18
Arrival-first viable states	0	0	12	15	18
Total phase-charge states	9	12	15	18	21

A critical arrival-first window begins at phase 1. Its leading charge is 2 and the following 2 phases contribute net deficit 2. These terms give the exact threshold in the capacity certificate.

MODEL A.06

Continuous industrial work

A fabrication phase makes the replacement chip. The following installation phase consumes the chip and draws one additional stored unit.

Phase ledger

Phase	0	1
Charge	5	2
Arrival	6	1
Deficit	-1	1
Largest prefix deficit	0	1
Arrival-first candidate	6	2

Capacity certificate

The totals are charge 7 and arrival 7, giving $D = 0$. The full-start capacities are $M = 1$ for the net ledger and $M_{\mathrm{a}} = 6$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $6 \mid 1 \to 0 \to 1$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 5, a full phase-zero start completes 1 safe transitions before the guard fails at phase 1.

Viability profile

At capacity 6, the viable starting charges by phase are $[0, 6]$, $[1, 6]$.

The kernel contains 13 phase-charge states at that capacity, and 15 when capacity increases by one.

Capacity sensitivity

Capacity	4	5	6	7	8
Net-ledger viable states	9	11	13	15	17
Arrival-first viable states	0	0	13	15	17
Total phase-charge states	10	12	14	16	18

A critical arrival-first window begins at phase 0. Its leading charge is 5 and the following 1 phase contributes net deficit 1. These terms give the exact threshold in the capacity certificate.

MODEL A.07

Instantaneous large supply

Direct use of the large arrival can power fabrication in the net ledger. In the arrival-first ledger the full charge passes through capacity.

Phase ledger

Phase	0	1	2
Charge	6	0	2
Arrival	8	0	0
Deficit	-2	0	2
Largest prefix deficit	0	2	2
Arrival-first candidate	8	2	2

Capacity certificate

The totals are charge 8 and arrival 8, giving $D = 0$. The full-start capacities are $M = 2$ for the net ledger and $M_{\mathrm{a}} = 8$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $8 \mid 2 \to 2 \to 0 \to 2$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 7, a full phase-zero start completes 2 safe transitions before the guard fails at phase 2.

Viability profile

At capacity 8, the viable starting charges by phase are $[0, 8]$, $[2, 8]$, $[2, 8]$.

The kernel contains 23 phase-charge states at that capacity, and 26 when capacity increases by one.

Capacity sensitivity

Capacity	6	7	8	9	10
Net-ledger viable states	17	20	23	26	29
Arrival-first viable states	0	0	23	26	29
Total phase-charge states	21	24	27	30	33

A critical arrival-first window begins at phase 0. Its leading charge is 6 and the following 2 phases contribute net deficit 2. These terms give the exact threshold in the capacity certificate.

MODEL A.08

Two separated deficits

A five-event cycle alternates transport and conversion. Two zero-arrival charges create separate maximal windows.

Phase ledger

Phase	0	1	2	3	4
Charge	3	2	4	1	2
Arrival	0	6	0	3	3
Deficit	3	-4	4	-2	-1
Largest prefix deficit	3	0	4	0	2
Arrival-first candidate	3	6	4	3	5

Capacity certificate

The totals are charge 12 and arrival 12, giving $D = 0$. The full-start capacities are $M = 4$ for the net ledger and $M_{\mathrm{a}} = 6$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $6 \to 3 \mid 4 \to 0 \to 2 \to 3 \to 0 \to 4$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 5, a full phase-zero start completes 2 safe transitions before the guard fails at phase 2.

Viability profile

At capacity 6, the viable starting charges by phase are $[3, 6]$, $[0, 6]$, $[4, 6]$, $[0, 6]$, $[2, 6]$.

The kernel contains 26 phase-charge states at that capacity, and 31 when capacity increases by one.

Capacity sensitivity

Capacity	4	5	6	7	8
Net-ledger viable states	16	21	26	31	36
Arrival-first viable states	0	0	26	31	36
Total phase-charge states	25	30	35	40	45

A critical arrival-first window begins at phase 1. Its leading charge is 2 and the following 1 phase contributes net deficit 4. These terms give the exact threshold in the capacity certificate.

MODEL A.09

A capacity bottleneck at installation

The large installation charge receives its own pulse, but later cleanup still draws on what the battery retains.

Phase ledger

Phase	0	1	2	3	4
Charge	1	1	7	2	1
Arrival	4	0	9	0	0
Deficit	-3	1	-2	2	1
Largest prefix deficit	0	2	1	3	1
Arrival-first candidate	3	2	10	3	1

Capacity certificate

The totals are charge 12 and arrival 13, giving $D = -1$. The full-start capacities are $M = 3$ for the net ledger and $M_{\mathrm{a}} = 10$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $10 \to 9 \to 8 \mid 3 \to 1 \to 0 \to 3 \to 2 \to 3$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 9, a full phase-zero start completes 4 safe transitions before the guard fails at phase 4.

Viability profile

At capacity 10, the viable starting charges by phase are $[0, 10]$, $[2, 10]$, $[1, 10]$, $[3, 10]$, $[1, 10]$. The kernel contains 48 phase-charge states at that capacity, and 53 when capacity increases by one.

Capacity sensitivity

Capacity	8	9	10	11	12
Net-ledger viable states	38	43	48	53	58
Arrival-first viable states	0	0	48	53	58
Total phase-charge states	45	50	55	60	65

A critical arrival-first window begins at phase 2. Its leading charge is 7 and the following 2 phases contribute net deficit 3. These terms give the exact threshold in the capacity certificate.

MODEL A.10

Surplus and a late fabrication

A zero-charge exchange opens a cycle with strict total surplus. The fabrication phase arrives late in a deficit run.

Phase ledger

Phase	0	1	2	3	4
Charge	0	2	2	4	3
Arrival	4	1	1	0	8
Deficit	-4	1	1	4	-5
Largest prefix deficit	2	6	5	4	0
Arrival-first candidate	6	7	6	4	5

Capacity certificate

The totals are charge 11 and arrival 14, giving $D = -3$. The full-start capacities are $M = 6$ for the net ledger and $M_{\mathrm{a}} = 7$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $7 \mid 7 \to 5 \to 4 \to 0 \to 4 \to 7$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 6, a full phase-zero start completes 3 safe transitions before the guard fails at phase 3.

Viability profile

At capacity 7, the viable starting charges by phase are $[2, 7]$, $[6, 7]$, $[5, 7]$, $[4, 7]$, $[0, 7]$.

The kernel contains 23 phase-charge states at that capacity, and 28 when capacity increases by one.

Capacity sensitivity

Capacity	5	6	7	8	9
Net-ledger viable states	0	18	23	28	33
Arrival-first viable states	0	0	23	28	33
Total phase-charge states	30	35	40	45	50

A critical arrival-first window begins at phase 1. Its leading charge is 2 and the following 2 phases contribute net deficit 5. These terms give the exact threshold in the capacity certificate.

MODEL A.11

Local supply and positive deficit

Every phase has an arrival, and one phase has a shortfall. Repetition tests the global cost of that shortfall.

Phase ledger

Phase	0	1	2	3
Charge	2	4	1	3
Arrival	2	2	1	3
Deficit	0	2	0	0
Largest prefix deficit	2	2	2	2
Arrival-first candidate	4	4	3	5

Capacity certificate

Each period costs 10 and receives 8. After k periods, the charge deficit is $2k$. Any initial store C therefore permits at most $\lfloor C/2 \rfloor$ whole periods.

A full phase-zero battery at capacity 7 completes 9 steps before failure at phase 1. This is the result of the direct guarded update.

Viability profile

The kernel is empty at every finite capacity in both conventions. The columns of local deficits identify the short windows that add constraints even before the cumulative obstruction applies.

Capacity sensitivity

Capacity	3	4	5	6	7
Net-ledger viable states	0	0	0	0	0
Arrival-first viable states	0	0	0	0	0
Total phase-charge states	16	20	24	28	32

The table gives zero viable states in each capacity sample. The strict cumulative deficit extends the same conclusion to every larger finite capacity.

MODEL A.12

Large transient supply with drift

An early ten-unit pulse supports several events. The average ledger still loses usable charge over whole periods.

Phase ledger

Phase	0	1	2	3	4	5
Charge	1	2	4	4	1	2
Arrival	10	0	0	0	3	0
Deficit	-9	2	4	4	-2	2
Largest prefix deficit	1	10	8	4	1	3
Arrival-first candidate	11	10	8	4	4	3

Capacity certificate

Each period costs 14 and receives 13. After k periods, the charge deficit is $1k$. Any initial store C therefore permits at most $\lfloor C/1 \rfloor$ whole periods.

A full phase-zero battery at capacity 13 completes 21 steps before failure at phase 3. This is the result of the direct guarded update.

Viability profile

The kernel is empty at every finite capacity in both conventions. The columns of local deficits identify the short windows that add constraints even before the cumulative obstruction applies.

Capacity sensitivity

Capacity	9	10	11	12	13
Net-ledger viable states	0	0	0	0	0
Arrival-first viable states	0	0	0	0	0
Total phase-charge states	60	66	72	78	84

The table gives zero viable states in each capacity sample. The strict cumulative deficit extends the same conclusion to every larger finite capacity.

MODEL A.13

Alternating high and low arrivals

Three pairs of events share a repeated supply rhythm. The table retains all six named phases of the organizational round.

Phase ledger

Phase	0	1	2	3	4	5
Charge	2	3	2	3	2	3
Arrival	5	0	5	0	5	0
Deficit	-3	3	-3	3	-3	3
Largest prefix deficit	0	3	0	3	0	3
Arrival-first candidate	5	3	5	3	5	3

Capacity certificate

The totals are charge 15 and arrival 15, giving $D = 0$. The full-start capacities are $M = 3$ for the net ledger and $M_{\mathrm{a}} = 5$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $5 \mid 3 \to 0 \to 3 \to 0 \to 3 \to 0 \to 3$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 4, a full phase-zero start completes 1 safe transitions before the guard fails at phase 1.

Viability profile

At capacity 5, the viable starting charges by phase are $[0, 5]$, $[3, 5]$, $[0, 5]$, $[3, 5]$, $[0, 5]$, $[3, 5]$. The kernel contains 27 phase-charge states at that capacity, and 33 when capacity increases by one.

Capacity sensitivity

Capacity	3	4	5	6	7
Net-ledger viable states	15	21	27	33	39
Arrival-first viable states	0	0	27	33	39
Total phase-charge states	24	30	36	42	48

A critical arrival-first window begins at phase 0. Its leading charge is 2 and the following 1 phase contributes net deficit 3. These terms give the exact threshold in the capacity certificate.

MODEL A.14

Maintenance with idle phases

Idle phases permit arrivals without charging a repair. The event table still checks the same clock for waiting and work.

Phase ledger

Phase	0	1	2	3	4	5	6
Charge	0	0	5	0	1	2	0
Arrival	0	4	0	1	1	0	2
Deficit	0	-4	5	-1	0	2	-2
Largest prefix deficit	2	2	6	1	2	2	0
Arrival-first candidate	2	6	6	2	3	2	2

Capacity certificate

The totals are charge 8 and arrival 8, giving $D = 0$. The full-start capacities are $M = 6$ for the net ledger and $M_{\mathrm{a}} = 6$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $6 \to 6 \mid 6 \to 1 \to 2 \to 2 \to 0 \to 2 \to 2 \to 6$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 5, a full phase-zero start completes 5 safe transitions before the guard fails at phase 5.

Viability profile

At capacity 6, the viable starting charges by phase are $[2, 6]$, $[2, 6]$, $[6, 6]$, $[1, 6]$, $[2, 6]$, $[2, 6]$, $[0, 6]$.

The kernel contains 34 phase-charge states at that capacity, and 41 when capacity increases by one.

Capacity sensitivity

Capacity	4	5	6	7	8
Net-ledger viable states	0	0	34	41	48
Arrival-first viable states	0	0	34	41	48
Total phase-charge states	35	42	49	56	63

A critical arrival-first window begins at phase 1. Its leading charge is 0 and the following 4 phases contribute net deficit 6. These terms give the exact threshold in the capacity certificate.

MODEL A.15

A long burst of exergy debt

Four surplus phases precede four deficit phases. The same mean charge and supply generate a nonzero storage requirement.

Phase ledger

Phase	0	1	2	3	4	5	6	7
Charge	2	2	2	2	2	2	2	2
Arrival	4	4	4	4	0	0	0	0
Deficit	-2	-2	-2	-2	2	2	2	2
Largest prefix deficit	0	2	4	6	8	6	4	2
Arrival-first candidate	4	6	8	10	8	6	4	2

Capacity certificate

The totals are charge 16 and arrival 16, giving $D = 0$. The full-start capacities are $M = 8$ for the net ledger and $M_{\mathrm{a}} = 10$ for arrival-first accounting.

At the arrival-first threshold, the phase-zero energy trace is $10 \to 8 \to 8 \to 8 \mid 8 \to 6 \to 4 \to 2 \to 0 \to 2 \to 4 \to 6 \to 8$. The vertical bar, when present, separates the transient from the repeating part. The closing entry repeats the first state of the period.

At capacity 9, a full phase-zero start completes 7 safe transitions before the guard fails at phase 7.

Viability profile

At capacity 10, the viable starting charges by phase are $[0, 10]$, $[2, 10]$, $[4, 10]$, $[6, 10]$, $[8, 10]$, $[6, 10]$, $[4, 10]$, $[2, 10]$.

The kernel contains 56 phase-charge states at that capacity, and 64 when capacity increases by one.

Capacity sensitivity

Capacity	8	9	10	11	12
Net-ledger viable states	40	48	56	64	72
Arrival-first viable states	0	0	56	64	72
Total phase-charge states	72	80	88	96	104

A critical arrival-first window begins at phase 3. Its leading charge is 2 and the following 4 phases contribute net deficit 8. These terms give the exact threshold in the capacity certificate.

CHAPTER 38

Typed sources and attainable repair

These presentations use finite rules, a feedstock seed f, and a designated completion r. A source label on a rule denotes the supplement whose inclusion makes that rule available. The two components of a charge are separate budget coordinates. The trees add charges along every branch and collect the source labels used.

The Pareto calculation of Theorem 11.3 uses a product order on source sets and grades. A tree is retained when no tree using a subset of its sources is weakly cheaper in both coordinates, with at least one strict comparison. An entry using more sources can remain useful if it offers a cheaper grade. An entry using fewer sources can remain useful when a cheap route is unavailable at a smaller boundary.

Each sheet also evaluates the ungraded rule table at every source portfolio. Minimal deletion sets meet every minimal successful portfolio. This is an independent cut certificate for the supply architecture and follows from the finite hypergraph calculation in Chapter 24.

MODEL B.01

One helper at a typed interface

One typed helper permits conversion and then an endogenous completion. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(2,1)$	A
e_2	f, m	p	$(2,2)$	—
e_3	p	r	$(1,1)$	—

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(5,4)$	$e_3(e_2(f, e_1(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_3, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 1 out of 2.

The minimal portfolios are $\{A\}$. The sources common to every minimal successful portfolio are $\{A\}$. Minimal deletion sets are $\{A\}$.

At budget $(5,4)$, the minimal feasible portfolios are $\{A\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m, p, r\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.02

Two hosts with different currencies

The two hosts supply the same role with different energy and material costs. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(1, 5)$	A
e_2	f	m	$(5, 1)$	B
e_3	f, m	p	$(2, 1)$	—
e_4	p	r	$(0, 1)$	—

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(3, 7)$	$e_4(e_3(f, e_1(f)))$
$\{B\}$	$(7, 3)$	$e_4(e_3(f, e_2(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_3, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 3 out of 4.

The minimal portfolios are $\{A\}$, $\{B\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A, B\}$.

At budget $(5, 4)$, the minimal feasible portfolios are none. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	2	2	0
2	1	1	0

The endogenous shortage certificate is the complement of its reached facts, namely $\{m, p, r\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.03

Complementary machinery

The mixed tree uses two independent suppliers, and a third can deliver the joint precursor. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(2,2)$	A
e_2	f	t	$(1,3)$	B
e_3	m,t	p	$(1,1)$	—
e_4	p	r	$(2,0)$	—
e_5	f	p	$(6,2)$	C

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{C\}$	$(8,2)$	$e_4(e_5(f))$
$\{A,B\}$	$(6,6)$	$e_4(e_3(e_1(f),e_2(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_1, r_2, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 5 out of 8.

The minimal portfolios are $\{C\}$, $\{A,B\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A,C\}$, $\{B,C\}$.

At budget $(6,5)$, the minimal feasible portfolios are none. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	3	1	0
2	3	3	0
3	1	1	0

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r,t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.04

Endogenous high-charge fallback

An endogenous source gives structural closure, and two supplements change attained budgets. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(5,5)$	—
e_2	f	m	$(1,2)$	A
e_3	m	p	$(1,1)$	—
e_4	p	r	$(2,1)$	—
e_5	f	r	$(2,7)$	B

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\varnothing$	$(8,7)$	$e_4(e_3(e_1(f)))$
$\{A\}$	$(4,4)$	$e_4(e_3(e_2(f)))$
$\{B\}$	$(2,7)$	$e_5(f)$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_1, with subscripts giving first stage. The endogenous catalog produces $\{f, m, p, r\}$. Successful portfolios number 4 out of 4.

The minimal portfolios are $\varnothing$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are none.

At budget $(4,4)$, the minimal feasible portfolios are $\{A\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	1	0
1	2	2	1
2	1	1	1

The zero-source row distinguishes existence of an endogenous proof from its affordability at the queried vector budget.

MODEL B.05

Reconversion by a second host

One host makes a helper and the other transforms it. A direct endogenous completion uses a different charge tradeoff. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(2,1)$	A
e_2	m	t	$(2,0)$	B
e_3	t	p	$(1,1)$	—
e_4	p	r	$(1,1)$	—
e_5	f	t	$(5,5)$	C
e_6	m	r	$(2,6)$	—

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(4,7)$	$e_6(e_1(f))$
$\{C\}$	$(7,7)$	$e_4(e_3(e_5(f)))$
$\{A,B\}$	$(6,3)$	$e_4(e_3(e_2(e_1(f))))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_2, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 6 out of 8.

The minimal portfolios are $\{A\}$, $\{C\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A, C\}$.

At budget $(7,4)$, the minimal feasible portfolios are $\{A,B\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	3	2	0
2	3	3	1
3	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r,t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.06

A joint output replaces parallel proofs

A supplier packages an intermediate that otherwise requires two branches. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(2,1)$	A
e_2	f	t	$(1,2)$	B
e_3	m,t	p	$(3,2)$	—
e_4	f	p	$(4,4)$	C
e_5	p	r	$(1,1)$	—

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{C\}$	$(5,5)$	$e_5(e_4(f))$
$\{A,B\}$	$(7,6)$	$e_5(e_3(e_1(f),e_2(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_1, r_2, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 5 out of 8.

The minimal portfolios are $\{C\}$, $\{A,B\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A,C\}$, $\{B,C\}$.

At budget $(5,5)$, the minimal feasible portfolios are $\{C\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	3	1	1
2	3	3	2
3	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r,t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.07

Two uses of the same catalyst proof

The paired premise names two occurrences of one role. The additive tree can choose a different supplier for each branch. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(1, 3)$	A
e_2	f	m	$(4, 1)$	B
e_3	m, m	p	$(1, 0)$	—
e_4	p	r	$(1, 1)$	—

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(4, 7)$	$e_4(e_3(e_1(f), e_1(f)))$
$\{B\}$	$(10, 3)$	$e_4(e_3(e_2(f), e_2(f)))$
$\{A, B\}$	$(7, 5)$	$e_4(e_3(e_1(f), e_2(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_3, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 3 out of 4.

The minimal portfolios are $\{A\}$, $\{B\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A, B\}$.

At budget $(7, 5)$, the minimal feasible portfolios are $\{A, B\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	2	2	0
2	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m, p, r\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.08

An imported completion certificate

A direct delivered repair competes with two certificates for an intermediate component. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(2,2)$	—
e_2	m	t	$(2,1)$	A
e_3	m,t	r	$(1,2)$	—
e_4	f	r	$(8,1)$	B
e_5	f	t	$(5,0)$	C

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(7,7)$	$e_3(e_1(f), e_2(e_1(f)))$
$\{B\}$	$(8,1)$	$e_4(f)$
$\{C\}$	$(8,4)$	$e_3(e_1(f), e_5(f))$

Boundary audit

The full catalog produces f_0, m_1, r_1, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f, m\}$. Successful portfolios number 7 out of 8.

The minimal portfolios are $\{A\}$, $\{B\}$, $\{C\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A, B, C\}$.

At budget $(8,5)$, the minimal feasible portfolios are $\{B\}$, $\{C\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	3	3	2
2	3	3	3
3	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{r, t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.09

A dormant productive cycle

A zero-charge cycle needs an enabling source to have a finite rooted proof. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	m	t	$(0, 0)$	—
e_2	t	m	$(0, 0)$	—
e_3	t	r	$(2, 2)$	—
e_4	f	m	$(2, 1)$	A
e_5	f	t	$(1, 3)$	B

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(4, 3)$	$e_3(e_1(e_4(f)))$
$\{B\}$	$(3, 5)$	$e_3(e_5(f))$

Boundary audit

The full catalog produces f_0, m_1, r_2, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 3 out of 4.

The minimal portfolios are $\{A\}$, $\{B\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A, B\}$.

At budget $(4, 4)$, the minimal feasible portfolios are $\{A\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	2	2	1
2	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m, r, t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.10

Cascade with a mandatory source

Both routes to the precursor use the same named interface, at different charges. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(1,2)$	A
e_2	f	m	$(3,0)$	B
e_3	m	p	$(2,2)$	C
e_4	p	r	$(1,1)$	—
e_5	m	p	$(3,1)$	C

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A,C\}$	$(4,5)$	$e_4(e_3(e_1(f)))$
$\{A,C\}$	$(5,4)$	$e_4(e_5(e_1(f)))$
$\{B,C\}$	$(6,3)$	$e_4(e_3(e_2(f)))$
$\{B,C\}$	$(7,2)$	$e_4(e_5(e_2(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_3, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 3 out of 8.

The minimal portfolios are $\{A,C\}$, $\{B,C\}$. The sources common to every minimal successful portfolio are $\{C\}$. Minimal deletion sets are $\{C\}$, $\{A,B\}$.

At budget $(6,5)$, the minimal feasible portfolios are $\{A,C\}$, $\{B,C\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	3	0	0
2	3	2	2
3	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.11

Nested industrial provision

A staged material chain competes with an integrated supplier of the finished precursor. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	w	$(2,3)$	A
e_2	w	m	$(1,1)$	B
e_3	f, m	p	$(2,0)$	C
e_4	p	r	$(1,2)$	—
e_5	f	p	$(7,2)$	D

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{D\}$	$(8,4)$	$e_4(e_5(f))$
$\{A,B,C\}$	$(6,6)$	$e_4(e_3(f, e_2(e_1(f))))$

Boundary audit

The full catalog produces f_0, m_2, p_1, r_2, w_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 9 out of 16.

The minimal portfolios are $\{D\}$, $\{A,B,C\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A,D\}$, $\{B,D\}$, $\{C,D\}$.

At budget $(8,6)$, the minimal feasible portfolios are $\{D\}$, $\{A,B,C\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	4	1	1
2	6	3	3
3	4	4	4
4	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r,w\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.12

Supplied seed with a dormant branch

A nullary supply establishes one role, while a second disconnected cycle never acquires a seed. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	1	m	$(1,2)$	A
e_2	m,t	r	$(1,1)$	—
e_3	f	t	$(2,3)$	B
e_4	u	p	$(0,0)$	—
e_5	p	u	$(0,0)$	—
e_6	p	r	$(0,1)$	—

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A,B\}$	$(4,6)$	$e_2(e_1,e_3(f))$

Boundary audit

The full catalog produces f_0, m_1, r_2, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 1 out of 4.

The minimal portfolios are $\{A,B\}$. The sources common to every minimal successful portfolio are $\{A,B\}$. Minimal deletion sets are $\{A\}$, $\{B\}$.

At budget $(4,6)$, the minimal feasible portfolios are $\{A,B\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	2	0	0
2	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r,t,u\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.13

A strict portfolio expansion lowers one grade

Adding a source to an already sufficient portfolio reduces energy demand via a different root. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(3, 2)$	A
e_2	m	r	$(3, 2)$	—
e_3	m	t	$(1, 0)$	B
e_4	t	r	$(1, 1)$	—
e_5	f	r	$(2, 8)$	C

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A\}$	$(6, 4)$	$e_2(e_1(f))$
$\{C\}$	$(2, 8)$	$e_5(f)$
$\{A, B\}$	$(5, 3)$	$e_4(e_3(e_1(f)))$

Boundary audit

The full catalog produces f_0, m_1, r_1, t_2, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 6 out of 8.

The minimal portfolios are $\{A\}$, $\{C\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A, C\}$.

At budget $(5, 4)$, the minimal feasible portfolios are $\{A, B\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	3	2	0
2	3	3	1
3	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m, r, t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.14

Separate suppliers for two joint paths

Three source-pair contracts have different prerequisite structure and different grade vectors. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(1,3)$	A
e_2	f	t	$(2,1)$	B
e_3	m,t	r	$(1,1)$	—
e_4	m	p	$(1,1)$	C
e_5	p	r	$(2,0)$	—
e_6	t	r	$(5,1)$	D

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\{A,B\}$	$(4,5)$	$e_3(e_1(f),e_2(f))$
$\{A,C\}$	$(4,4)$	$e_5(e_4(e_1(f)))$
$\{B,D\}$	$(7,2)$	$e_6(e_2(f))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_2, t_1, with subscripts giving first stage. The endogenous catalog produces $\{f\}$. Successful portfolios number 8 out of 16.

The minimal portfolios are $\{A,B\}$, $\{A,C\}$, $\{B,D\}$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are $\{A,B\}$, $\{A,D\}$, $\{B,C\}$.

At budget $(5,5)$, the minimal feasible portfolios are $\{A,B\}$, $\{A,C\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	0	0
1	4	0	0
2	6	3	2
3	4	4	3
4	1	1	1

The endogenous shortage certificate is the complement of its reached facts, namely $\{m,p,r,t\}$. It includes the target and each producing rule with output in this set has a premise in it.

MODEL B.15

A low-charge supplement with a higher material cost

Several additions preserve the endogenous proof while introducing tradeoffs among attainable charges. The seed f is free and the target r is a completed repair.

Primitive rule catalog

Rule	Premise	Output	Charge	Source
e_1	f	m	$(2,2)$	—
e_2	m	p	$(2,2)$	—
e_3	p	r	$(2,2)$	—
e_4	m	p	$(0,4)$	A
e_5	p	r	$(1,1)$	B
e_6	f	r	$(1,9)$	C

Minimal source and charge entries for the target

Source set	Charge	Representative derivation
$\varnothing$	$(6,6)$	$e_3(e_2(e_1(f)))$
$\{A\}$	$(4,8)$	$e_3(e_4(e_1(f)))$
$\{B\}$	$(5,5)$	$e_5(e_2(e_1(f)))$
$\{C\}$	$(1,9)$	$e_6(f)$
$\{A,B\}$	$(3,7)$	$e_5(e_4(e_1(f)))$

Boundary audit

The full catalog produces f_0, m_1, p_2, r_1, with subscripts giving first stage. The endogenous catalog produces $\{f,m,p,r\}$. Successful portfolios number 8 out of 8.

The minimal portfolios are $\varnothing$. The sources common to every minimal successful portfolio are $\varnothing$. Minimal deletion sets are none.

At budget $(4,7)$, the minimal feasible portfolios are $\{A,B\}$. The attained comparison uses both grade coordinates of a single derivation.

Portfolio counts by size

Number of named sources	Portfolios	Closed	Within budget
0	1	1	0
1	3	3	0
2	3	3	1
3	1	1	1

The zero-source row distinguishes existence of an endogenous proof from its affordability at the queried vector budget.

CHAPTER 39

Possible, robust, and stochastic repair

Each small graph in this family has a finite set of named control choices. Once the control is selected, its probability distribution gives the possible next states. A dagger denotes an unsafe state. A dash in a control column denotes absence of that control.

Existential viability chooses a possible successor; robust viability asks for a control whose whole support stays safe. The two deletion calculations thus use different predecessor operators on the same state set. A finite rank j removes a state on the j-th application. Infinite rank denotes a kernel member. By Proposition 15.3, the robust kernel also gives the probability-one states.

The rational entries q_j are independently computed by dynamic programming for the stated finite horizons. Together with the lasso or fixed control certificate, these tables distinguish the existence of an execution from a guarantee against every supported outcome.

MODEL C.01

A reliable control and a tempting shortcut

At the initial state, one control repeats safely and the other enters a faster but exposed route.

Control table

State	Control 1	Control 2	Control 3
0	$\{0{:}1\}$	$\{1{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—
1	$\{2{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—
2	$\{2{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—
3	$\{3{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	∞	1	1	1	1
1	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$
2	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$
3	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3\}$. The robust and almost-sure kernel is $\{0, 3\}$.
A fixed safe control table, as pairs of state and control number, is $(0, 1)$, $(3, 1)$.
A possible path and repeating loop from state zero are $0 \to 0$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.02

A fragile loop

Each loop attempt at state zero carries a positive failure probability. A separate deterministic pair survives.

Control table

State	Control 1	Control 2	Control 3
0	$\{0{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	—	—
1	$\{1{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—
2	$\{3{:}1\}$	—	—
3	$\{2{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
1	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3\}$. The robust and almost-sure kernel is $\{2, 3\}$.

A fixed safe control table, as pairs of state and control number, is $(2, 1)$, $(3, 1)$.

A possible path and repeating loop from state zero are $0 \to 0$.

The best probability at horizon five from state zero is $\frac{32}{243}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.03

Preparation before entering a stable regime

Every successor of the first control can reach the same stable state within two more steps.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}\frac{2}{3}, 2{:}\frac{1}{3}\}$	—	—
1	$\{2{:}1\}$	—	—
2	$\{3{:}1\}$	—	—
3	$\{3{:}1\}$	—	—
4	$\{4{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	∞	1	1	1	1
1	∞	∞	1	1	1	1
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1
4	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 4\}$. The robust and almost-sure kernel is $\{0, 1, 2, 3\}$. A fixed safe control table, as pairs of state and control number, is $(0, 1)$, $(1, 1)$, $(2, 1)$, $(3, 1)$. A possible path and repeating loop from state zero are $0 \to 1 \to 2 \mid 3 \to 3$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.04

Choosing among finite route lengths

Every path eventually reaches a dead end. Choosing the earliest state on the chain gives a longer finite horizon.

Control table

State	Control 1	Control 2	Control 3
0	{1:1}	{2:1}	{3:1}
1	{2:1}	—	—
2	{3:1}	—	—
3	{4:1}	—	—
4	{†:1}	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	5	5	1	1	1	0
1	4	4	1	1	1	0
2	3	3	1	1	0	0
3	2	2	1	0	0	0
4	1	1	0	0	0	0

Kernel certificate

The possible-continuation kernel is $\varnothing$. The robust and almost-sure kernel is $\varnothing$.

Every state has a finite robust deletion rank. Each rank is a certificate of a finite failure horizon.

State zero has at most 4 possible further safe steps.

The best probability at horizon five from state zero is 0.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	{1, 2, 3}	{1, 2}	{1, 2, 3}
1	{1}	{1}	{1}
2	{1}	{1}	{1}
3	{1}	{1}	{1}
4	{1}	{1}	{1}

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.05

Local repair with a risky extra mode

The safe route reaches a stable two-state cycle. The other controls supply several finite-risk continuations.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	$\{2{:}1\}$	—
1	$\{1{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	—	—
2	$\{3{:}1\}$	$\{4{:}1\}$	—
3	$\{2{:}1\}$	—	—
4	$\{4{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	∞	1	1	1	1
1	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1
4	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 4\}$. The robust and almost-sure kernel is $\{0, 2, 3\}$.
A fixed safe control table, as pairs of state and control number, is $(0, 2)$, $(2, 1)$, $(3, 1)$.
A possible path and repeating loop from state zero are $0 \mid 1 \to 1$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{2\}$	$\{2\}$	$\{2\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1, 2\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.06

A disturbance prevents commitment

The first control can send the process to either a fragile loop or a deterministic tail.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}\frac{1}{2}, 2{:}\frac{1}{2}\}$	—	—
1	$\{1{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—
2	$\{3{:}1\}$	—	—
3	$\{3{:}1\}$	—	—
4	$\{4{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	2	1	$\frac{3}{4}$	$\frac{5}{8}$	$\frac{17}{32}$
1	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1
4	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 4\}$. The robust and almost-sure kernel is $\{2, 3, 4\}$.
A fixed safe control table, as pairs of state and control number, is $(2, 1)$, $(3, 1)$, $(4, 1)$.
A possible path and repeating loop from state zero are $0 \mid 1 \to 1$.
The best probability at horizon five from state zero is $\frac{17}{32}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.07

Finite delivery and two failure ranks

An initial delivery leaves two continuation routes, with distinct support and horizon calculations.

Control table

State	Control 1	Control 2	Control 3
0	$\{1:\frac{1}{2}, \dagger:\frac{1}{2}\}$	$\{2:1\}$	—
1	$\{3:1\}$	—	—
2	$\{2:\frac{1}{3}, 3:\frac{1}{3}, \dagger:\frac{1}{3}\}$	—	—
3	$\{\dagger:1\}$	—	—
4	$\{4:1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	2	1	$\frac{2}{3}$	$\frac{2}{9}$	$\frac{2}{81}$
1	2	2	1	0	0	0
2	∞	1	$\frac{2}{3}$	$\frac{2}{9}$	$\frac{2}{27}$	$\frac{2}{243}$
3	1	1	0	0	0	0
4	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 2, 4\}$. The robust and almost-sure kernel is $\{4\}$.

A fixed safe control table, as pairs of state and control number, is $(4, 1)$.

A possible path and repeating loop from state zero are $0 \mid 2 \to 2$.

The best probability at horizon five from state zero is $\frac{2}{81}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{2\}$	$\{2\}$	$\{2\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.08

A restart from a safe intermediate

A choice at state two permits a restart to the initial process. The safe support includes a nondeterministic branch.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}1\}$	—	—
1	$\{0{:}\frac{1}{2}, 2{:}\frac{1}{2}\}$	—	—
2	$\{2{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	$\{3{:}1\}$	—
3	$\{0{:}\frac{1}{2}, 3{:}\frac{1}{2}\}$	—	—
4	$\{\dagger{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	∞	1	1	1	1
1	∞	∞	1	1	1	1
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1
4	1	1	0	0	0	0

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3\}$. The robust and almost-sure kernel is $\{0, 1, 2, 3\}$. A fixed safe control table, as pairs of state and control number, is $(0, 1)$, $(1, 1)$, $(2, 2)$, $(3, 1)$. A possible path and repeating loop from state zero are $0 \to 1 \to 0$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{2\}$	$\{2\}$	$\{2\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.09

A split into two independent robust basins

The initial transition selects one of two basins. Every branch has a safe control at every stage.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}\frac{1}{2}, 2{:}\frac{1}{2}\}$	—	—
1	$\{1{:}\frac{1}{2}, 3{:}\frac{1}{2}\}$	—	—
2	$\{2{:}\frac{1}{2}, 4{:}\frac{1}{2}\}$	—	—
3	$\{3{:}1\}$	—	—
4	$\{4{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	∞	1	1	1	1
1	∞	∞	1	1	1	1
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1
4	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 4\}$. The robust and almost-sure kernel is $\{0, 1, 2, 3, 4\}$.

A fixed safe control table, as pairs of state and control number, is $(0, 1)$, $(1, 1)$, $(2, 1)$, $(3, 1)$, $(4, 1)$.

A possible path and repeating loop from state zero are $0 \mid 1 \to 1$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.10

Failure under every mixed control

Each of the first three states has a possible loop and every available control risks failure.

Control table

State	Control 1	Control 2	Control 3
0	$\{0{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	$\{0{:}\frac{1}{3}, 1{:}\frac{1}{3}, \dagger{:}\frac{1}{3}\}$	—
1	$\{1{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	$\{1{:}\frac{1}{3}, 2{:}\frac{1}{3}, \dagger{:}\frac{1}{3}\}$	—
2	$\{0{:}\frac{1}{3}, 2{:}\frac{1}{3}, \dagger{:}\frac{1}{3}\}$	—	—
3	$\{3{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
1	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
2	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
3	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3\}$. The robust and almost-sure kernel is $\{3\}$.
A fixed safe control table, as pairs of state and control number, is $(3, 1)$.
A possible path and repeating loop from state zero are $0 \to 0$.
The best probability at horizon five from state zero is $\frac{32}{243}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1, 2\}$	$\{1, 2\}$	$\{1, 2\}$
1	$\{1, 2\}$	$\{1, 2\}$	$\{1, 2\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.11

Long deterministic prefix before a fragile loop

A lasso in the possible graph is preceded by a long preparation chain. The first possible failure is delayed by that prefix.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}1\}$	—	—
1	$\{2{:}1\}$	—	—
2	$\{3{:}1\}$	—	—
3	$\{4{:}1\}$	—	—
4	$\{4{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	5	1	1	1	$\frac{2}{3}$
1	∞	4	1	1	1	$\frac{4}{9}$
2	∞	3	1	1	$\frac{2}{3}$	$\frac{8}{27}$
3	∞	2	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{16}{81}$
4	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 4\}$. The robust and almost-sure kernel is $\varnothing$.
Every state has a finite robust deletion rank. Each rank is a certificate of a finite failure horizon.
A possible path and repeating loop from state zero are $0 \to 1 \to 2 \to 3 \mid 4 \to 4$.
The best probability at horizon five from state zero is $\frac{2}{3}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.12

Control order on two exposed components

Each preparatory choice has one unsafe outcome, although either successful outcome reaches the same robust terminal mode.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	$\{2{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—
1	$\{3{:}1\}$	—	—
2	$\{3{:}1\}$	—	—
3	$\{3{:}1\}$	—	—
4	$\{4{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	1	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$
1	∞	∞	1	1	1	1
2	∞	∞	1	1	1	1
3	∞	∞	1	1	1	1
4	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 4\}$. The robust and almost-sure kernel is $\{1, 2, 3, 4\}$.
A fixed safe control table, as pairs of state and control number, is $(1, 1)$, $(2, 1)$, $(3, 1)$, $(4, 1)$.
A possible path and repeating loop from state zero are $0 \to 1 \mid 3 \to 3$.
The best probability at horizon five from state zero is $\frac{1}{2}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1, 2\}$	$\{1, 2\}$	$\{1, 2\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.13

A renewal choice at the bottleneck

Only one initial route avoids immediate risk. Its successor still admits an exposed branch under every control.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}1\}$	$\{2{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—
1	$\{0{:}\frac{1}{2}, 2{:}\frac{1}{2}\}$	$\{3{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—
2	$\{2{:}\frac{1}{2}, \dagger{:}\frac{1}{2}\}$	—	—
3	$\{3{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	3	1	1	$\frac{3}{4}$	$\frac{1}{2}$
1	∞	2	1	$\frac{3}{4}$	$\frac{5}{8}$	$\frac{1}{2}$
2	∞	1	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{32}$
3	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3\}$. The robust and almost-sure kernel is $\{3\}$.
A fixed safe control table, as pairs of state and control number, is $(3, 1)$.
A possible path and repeating loop from state zero are $0 \to 1 \to 0$.
The best probability at horizon five from state zero is $\frac{1}{2}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{2\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.14

Nested rare failures

Failure is reached through successive rare transitions. The probability-one horizon can stay large before robust deletion reaches the first state.

Control table

State	Control 1	Control 2	Control 3
0	$\{0{:}\frac{2}{3}, 1{:}\frac{1}{3}\}$	—	—
1	$\{1{:}\frac{2}{3}, 2{:}\frac{1}{3}\}$	—	—
2	$\{2{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	—	—
3	$\{3{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	3	1	1	$\frac{26}{27}$	$\frac{64}{81}$
1	∞	2	1	$\frac{8}{9}$	$\frac{20}{27}$	$\frac{112}{243}$
2	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
3	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3\}$. The robust and almost-sure kernel is $\{3\}$.

A fixed safe control table, as pairs of state and control number, is $(3, 1)$.

A possible path and repeating loop from state zero are $0 \to 0$.

The best probability at horizon five from state zero is $\frac{64}{81}$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

MODEL C.15

An absorbing output and an unavailable control

State four is safe but has no admitted action. The reachable states divide into a robust delivery branch and a fragile branch.

Control table

State	Control 1	Control 2	Control 3
0	$\{1{:}1\}$	$\{2{:}1\}$	—
1	$\{1{:}\frac{1}{2}, 3{:}\frac{1}{2}\}$	—	—
2	$\{2{:}\frac{2}{3}, \dagger{:}\frac{1}{3}\}$	—	—
3	$\{3{:}1\}$	—	—
4	—	—	—
5	$\{5{:}1\}$	—	—

An entry $y : p$ gives successor y with probability p. Each displayed control has total probability one.

Deletion ranks and optimal finite-horizon probabilities

State	Possible rank	Robust rank	q_1	q_2	q_3	q_5
0	∞	∞	1	1	1	1
1	∞	∞	1	1	1	1
2	∞	1	$\frac{2}{3}$	$\frac{4}{9}$	$\frac{8}{27}$	$\frac{32}{243}$
3	∞	∞	1	1	1	1
4	1	1	0	0	0	0
5	∞	∞	1	1	1	1

Kernel certificate

The possible-continuation kernel is $\{0, 1, 2, 3, 5\}$. The robust and almost-sure kernel is $\{0, 1, 3, 5\}$. A fixed safe control table, as pairs of state and control number, is $(0, 1)$, $(1, 1)$, $(3, 1)$, $(5, 1)$. A possible path and repeating loop from state zero are $0 \mid 1 \to 1$.

First controls attaining the horizon optima

State	At horizon 1	At horizon 3	At horizon 5
0	$\{1, 2\}$	$\{1\}$	$\{1\}$
1	$\{1\}$	$\{1\}$	$\{1\}$
2	$\{1\}$	$\{1\}$	$\{1\}$
3	$\{1\}$	$\{1\}$	$\{1\}$
4	$\varnothing$	$\varnothing$	$\varnothing$
5	$\{1\}$	$\{1\}$	$\{1\}$

After the first outcome, the optimizer uses the continuation value at the remaining horizon.

CHAPTER 40

Selectable maintenance routes

The fourth family allows the controller to choose among several ledgered transitions. Every state and edge is labeled explicitly. The battery uses arrival-first accounting and state zero is the specified full-battery start. The edge deficit is the charge less the arrival.

Two independent computations give the threshold tables. One expands the states into pairs of base state and charge and prunes the graph. The other represents the viable charge at each base state by an upper interval and uses the backward threshold recursion of Chapter 29. The cardinalities are taken in the expanded graph.

For a graph with n base states and maximum charge $c_{\max}$, a reachable nonpositive simple cycle yields a viable capacity at most $2nc_{\max}$. Take a simple path of length at most $n-1$ to the cycle and then repeat its at most n edges. In the arrival-first formula, every window is bounded by removing complete nonpositive periods and summing charges on at most $2n-1$ edges. This provides a finite search bound for each threshold. The cycle-mean comparison verifies whether a finite threshold exists. Each specified cut removes one named edge.

MODEL D.01

Two repair routes with equal throughput

An even route and a pulsed route accomplish the same task with different storage. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	4	2	-2
2	1	0	0	2	2
3	0	2	2	2	0
4	2	0	2	2	0

Charge thresholds in the lifted graph

Capacity	0	1	2	3	4
Minimum at state 0	∞	∞	0	0	0
Minimum at state 1	∞	∞	2	2	2
Minimum at state 2	∞	∞	0	0	0
Kernel cardinality	0	0	7	10	13

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 2. Removing edge 4 changes the mean to 0 and the full-start threshold to 4.

A lifted lasso (base state, stored exergy) is $(0, 2) \mid (2, 0) \to (0, 0) \to (2, 0)$.

At capacity 1, the full initial state has deletion rank 1, so it completes at most 0 transitions.

Minimum starting charges at capacity 2

Boundary	State 0	State 1	State 2	States in kernel
Full	0	2	0	7
Without edge 4	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.02

A cheap loop beyond a costly entrance

A costly prefix precedes a balanced persistent loop. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	0	5	5
2	1	2	1	3	2
3	2	2	2	2	0
4	0	0	1	2	1

Charge thresholds in the lifted graph

Capacity	5	6	7	8	9
Minimum at state 0	∞	∞	7	7	7
Minimum at state 1	2	2	2	2	2
Minimum at state 2	0	0	0	0	0
Kernel cardinality	10	12	15	18	21

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 7. Removing edge 3 changes the mean to 1 and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0, 7) \to (1, 2) \mid (2, 0) \to (2, 0)$.

At capacity 6, the full initial state has deletion rank 6, so it completes at most 5 transitions.

Minimum starting charges at capacity 7

Boundary	State 0	State 1	State 2	States in kernel
Full	7	2	0	15
Without edge 3	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.03

One mixed branch restores net supply

A host contribution repays a debt incurred during preparation. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	1	3	2
2	1	0	4	2	-2
3	1	2	0	1	1
4	2	0	2	2	0
5	0	0	0	1	1

Charge thresholds in the lifted graph

Capacity	2	3	4	5	6
Minimum at state 0	∞	∞	2	2	2
Minimum at state 1	∞	∞	0	0	0
Minimum at state 2	∞	∞	2	2	2
Kernel cardinality	0	0	11	14	17

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 4. Removing edge 2 changes the mean to 1 and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0,4) \to (0,3) \mid (0,2) \to (1,0) \to (0,2)$.

At capacity 3, the full initial state has deletion rank 5, so it completes at most 4 transitions.

Minimum starting charges at capacity 4

Boundary	State 0	State 1	State 2	States in kernel
Full	2	0	2	11
Without edge 2	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.04

Two independent sinks

Two accessible cycles carry different charges and startup costs. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	0	3	3
2	0	2	1	1	0
3	1	1	3	3	0
4	2	3	1	2	1
5	3	2	3	2	-1

Charge thresholds in the lifted graph

Capacity	1	2	3	4	5
Minimum at state 0	∞	∞	1	1	1
Minimum at state 1	∞	∞	0	0	0
Minimum at state 2	∞	∞	1	1	1
Minimum at state 3	∞	∞	0	0	0
Kernel cardinality	0	0	14	18	22

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 3. Removing edge 5 changes the mean to 0 and the full-start threshold to 3.

A lifted lasso (base state, stored exergy) is $(0, 3) \mid (1, 0) \to (1, 0)$.

At capacity 2, the full initial state has deletion rank 4, so it completes at most 3 transitions.

Minimum starting charges at capacity 3

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	1	0	1	0	14
Without edge 5	3	0	∞	∞	5

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.05

A future surplus behind a long prefix

Abundant surplus appears only after three debt-bearing preparatory transitions. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	0	3	3
2	1	2	0	2	2
3	2	3	0	2	2
4	3	3	4	1	-3
5	0	0	1	2	1

Charge thresholds in the lifted graph

Capacity	5	6	7	8	9
Minimum at state 0	∞	∞	7	7	7
Minimum at state 1	4	4	4	4	4
Minimum at state 2	2	2	2	2	2
Minimum at state 3	0	0	0	0	0
Kernel cardinality	12	15	19	23	27

Lasso and boundary comparison

The minimum reachable mean deficit is -3. The full-start threshold is 7. Removing edge 4 changes the mean to 1 and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0, 7) \to (1, 4) \to (2, 2) \to (3, 0) \to (3, 3) \mid (3, 6) \to (3, 6)$.

At capacity 6, the full initial state has deletion rank 6, so it completes at most 5 transitions.

Minimum starting charges at capacity 7

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	7	4	2	0	19
Without edge 4	∞	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.06

A control that discards excess supply

Large pulses can be clipped before later uses. A route with fewer phases changes the capacity requirement. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	10	4	-6
2	1	2	0	4	4
3	2	0	0	2	2
4	0	2	4	2	-2
5	2	2	2	3	1

Charge thresholds in the lifted graph

Capacity	2	3	4	5	6
Minimum at state 0	∞	∞	0	0	0
Minimum at state 1	∞	∞	∞	∞	6
Minimum at state 2	∞	∞	2	2	2
Kernel cardinality	0	0	8	10	13

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 4. Removing edge 4 changes the mean to 0 and the full-start threshold to 10.

A lifted lasso (base state, stored exergy) is $(0, 4) \mid (2, 2) \to (0, 0) \to (2, 2)$.

At capacity 3, the full initial state has deletion rank 3, so it completes at most 2 transitions.

Minimum starting charges at capacity 4

Boundary	State 0	State 1	State 2	States in kernel
Full	0	∞	2	8
Without edge 4	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.07

An unreachable viable component

One pair is self-maintaining, and the specified start reaches only the positive-deficit pair. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	0	1	1
2	1	0	1	2	1
3	2	3	2	1	-1
4	3	2	2	1	-1
5	1	1	1	3	2

Charge thresholds in the lifted graph

Capacity	1	2	3	4	5
Minimum at state 0	∞	∞	∞	∞	∞
Minimum at state 1	∞	∞	∞	∞	∞
Minimum at state 2	0	0	0	0	0
Minimum at state 3	0	0	0	0	0
Kernel cardinality	4	6	8	10	12

Lasso and boundary comparison

The minimum reachable mean deficit is 1. The full-start threshold is ∞. Removing edge 5 changes the mean to 1 and the full-start threshold to ∞.

The reachable graph has no cycle with nonpositive deficit. The positive-drift obstruction gives a finite horizon for every finite battery.

Minimum starting charges at capacity 3

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	∞	∞	0	0	8
Without edge 5	∞	∞	0	0	8

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.08

A zero-charge route with a late cost

A free transport and storage phase competes with a local charge-one loop. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	0	0	0
2	1	2	3	0	-3
3	2	0	0	3	3
4	0	0	1	1	0
5	1	1	2	0	-2

Charge thresholds in the lifted graph

Capacity	0	1	2
Minimum at state 0	0	0	0
Minimum at state 1	0	0	0
Minimum at state 2	∞	∞	∞
Kernel cardinality	2	4	6

Lasso and boundary comparison

The minimum reachable mean deficit is -2. The full-start threshold is 0. Removing edge 4 changes the mean to -2 and the full-start threshold to 0.

A lifted lasso (base state, stored exergy) is $(0,0) \mid (1,0) \to (1,0)$.

Minimum starting charges at capacity 0

Boundary	State 0	State 1	State 2	States in kernel
Full	0	0	∞	2
Without edge 4	0	0	∞	2

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.09

Shared return through the interface

Both routes use the same return from the common assembly state. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	4	3	-1
2	1	2	0	1	1
3	2	0	3	3	0
4	0	3	2	2	0
5	3	2	2	1	-1
6	1	1	0	1	1

Charge thresholds in the lifted graph

Capacity	1	2	3	4	5
Minimum at state 0	∞	∞	0	0	0
Minimum at state 1	∞	∞	1	1	1
Minimum at state 2	∞	∞	0	0	0
Minimum at state 3	∞	∞	0	0	0
Kernel cardinality	0	0	15	19	23

Lasso and boundary comparison

The minimum reachable mean deficit is $\frac{-1}{3}$. The full-start threshold is 3. Removing edge 3 changes the mean to 1 and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0, 3) \to (3, 1) \to (2, 2) \mid (0, 0) \to (3, 0) \to (2, 1) \to (0, 0)$.

At capacity 2, the full initial state has deletion rank 3, so it completes at most 2 transitions.

Minimum starting charges at capacity 3

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	0	1	0	0	15
Without edge 3	∞	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.10

A reserve created by postponing work

Waiting on the second route accumulates energy before rejoining the repair phases. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	3	1	-2
2	1	0	0	2	2
3	1	2	5	0	-5
4	2	1	0	1	1
5	2	0	0	3	3

Charge thresholds in the lifted graph

Capacity	0	1	2	3
Minimum at state 0	∞	0	0	0
Minimum at state 1	∞	0	0	0
Minimum at state 2	∞	1	1	1
Kernel cardinality	0	5	8	11

Lasso and boundary comparison

The minimum reachable mean deficit is -2. The full-start threshold is 1. Removing edge 3 changes the mean to 0 and the full-start threshold to 3.

A lifted lasso (base state, stored exergy) is $(0, 1) \mid (1, 0) \to (2, 1) \to (1, 0)$.

At capacity 0, the full initial state has deletion rank 1, so it completes at most 0 transitions.

Minimum starting charges at capacity 1

Boundary	State 0	State 1	State 2	States in kernel
Full	0	0	1	5
Without edge 3	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.11

An irreversible mode switch

The first mode loses exergy at every loop. Entering the sustainable mode has a one-time startup cost. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	0	1	2	1
2	0	1	0	2	2
3	1	2	2	1	-1
4	2	1	2	1	-1
5	2	3	0	4	4
6	3	3	3	1	-2

Charge thresholds in the lifted graph

Capacity	0	1	2	3	4
Minimum at state 0	∞	∞	2	2	2
Minimum at state 1	∞	0	0	0	0
Minimum at state 2	∞	0	0	0	0
Minimum at state 3	∞	0	0	0	0
Kernel cardinality	0	6	10	14	18

Lasso and boundary comparison

The minimum reachable mean deficit is -2. The full-start threshold is 2. Removing edge 2 changes the mean to 1 and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0, 2) \to (1, 0) \mid (2, 1) \to (1, 1) \to (2, 1)$.

At capacity 1, the full initial state has deletion rank 1, so it completes at most 0 transitions.

Minimum starting charges at capacity 2

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	2	0	0	0	10
Without edge 2	∞	0	0	0	9

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.12

A cycle with one high-cost joint event

A high-charge joint event competes with a distributed alternative. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	2	2	0
2	1	0	8	6	-2
3	0	2	3	3	0
4	2	3	3	3	0
5	3	0	2	2	0

Charge thresholds in the lifted graph

Capacity	1	2	3	4	5
Minimum at state 0	∞	∞	0	0	0
Minimum at state 1	∞	∞	∞	∞	∞
Minimum at state 2	∞	∞	0	0	0
Minimum at state 3	∞	∞	0	0	0
Kernel cardinality	0	0	12	15	18

Lasso and boundary comparison

The minimum reachable mean deficit is -1. The full-start threshold is 3. Removing edge 3 changes the mean to -1 and the full-start threshold to 6.

A lifted lasso (base state, stored exergy) is $(0, 3) \mid (2, 0) \to (3, 0) \to (0, 0) \to (2, 0)$.

At capacity 2, the full initial state has deletion rank 2, so it completes at most 1 transitions.

Minimum starting charges at capacity 3

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	0	∞	0	0	12
Without edge 3	∞	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.13

Dead-end fabrication

The inexpensive prefix to fabrication ends at a state with no outgoing edge. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	2	1	-1
2	1	2	4	3	-1
3	0	3	1	2	1
4	3	0	3	2	-1

Charge thresholds in the lifted graph

Capacity	1	2	3	4	5
Minimum at state 0	∞	∞	1	1	1
Minimum at state 1	∞	∞	∞	∞	∞
Minimum at state 2	∞	∞	∞	∞	∞
Minimum at state 3	∞	∞	0	0	0
Kernel cardinality	0	0	7	9	11

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 3. Removing edge 4 changes the mean to ∞ and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0, 3) \to (3, 1) \mid (0, 1) \to (3, 0) \to (0, 1)$.

At capacity 2, the full initial state has deletion rank 4, so it completes at most 3 transitions.

Minimum starting charges at capacity 3

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	1	∞	∞	0	7
Without edge 4	∞	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.14

Extra supply without a change of cycle type

Two transitions share endpoints and charge, while only one provides a useful pulse. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	0	3	3
2	0	1	4	3	-1
3	1	2	0	2	2
4	2	0	3	2	-1
5	1	1	0	1	1

Charge thresholds in the lifted graph

Capacity	3	4	5	6	7
Minimum at state 0	∞	∞	1	1	1
Minimum at state 1	∞	∞	2	2	2
Minimum at state 2	∞	∞	0	0	0
Kernel cardinality	0	0	15	18	21

Lasso and boundary comparison

The minimum reachable mean deficit is 0. The full-start threshold is 5. Removing edge 2 changes the mean to 1 and the full-start threshold to ∞.

A lifted lasso (base state, stored exergy) is $(0, 5) \mid (1, 2) \to (2, 0) \to (0, 1) \to (1, 2)$.

At capacity 4, the full initial state has deletion rank 3, so it completes at most 2 transitions.

Minimum starting charges at capacity 5

Boundary	State 0	State 1	State 2	States in kernel
Full	1	2	0	15
Without edge 2	∞	∞	∞	0

The boundary comparison keeps the battery and the state catalog fixed.

MODEL D.15

Competing return paths under a common ledger

The routes share a startup transition and differ in their internal charge timing. The initial base state is zero and the battery starts full.

Competing transitions

Edge	From	To	Arrival	Charge	Deficit
1	0	1	4	4	0
2	1	2	2	1	-1
3	2	0	0	1	1
4	1	3	0	3	3
5	3	0	5	2	-3
6	2	3	3	2	-1

Charge thresholds in the lifted graph

Capacity	2	3	4	5	6
Minimum at state 0	∞	∞	0	0	0
Minimum at state 1	∞	∞	0	0	0
Minimum at state 2	∞	∞	0	0	0
Minimum at state 3	∞	∞	0	0	0
Kernel cardinality	0	0	20	24	28

Lasso and boundary comparison

The minimum reachable mean deficit is $\frac{-5}{4}$. The full-start threshold is 4. Removing edge 3 changes the mean to $\frac{-5}{4}$ and the full-start threshold to 4.

A lifted lasso (base state, stored exergy) is $(0, 4) \mid (1, 0) \to (2, 1) \to (0, 0) \to (1, 0)$.

At capacity 3, the full initial state has deletion rank 1, so it completes at most 0 transitions.

Minimum starting charges at capacity 4

Boundary	State 0	State 1	State 2	State 3	States in kernel
Full	0	0	0	0	20
Without edge 3	0	0	0	0	20

The boundary comparison keeps the battery and the state catalog fixed.

CHAPTER 41

Stock and regeneration schedules

The event words here retain token multiplicities. A label in both input and output is a catalytic use when the two multiplicities coincide. A word mmf requires two units of m and one of f. The partial stock updates use the enabling rule of Chapter 9.

Each sheet first computes the minimal stock for its original event sequence. It shows the direct execution from that minimum and tests sharpness by lowering every positive coordinate in turn. It then inserts one coupled event and computes the minimal stock and the peak used by the enlarged sequence.

For initial stock $q + u$, with minimum q and net change d, the largest number of complete rounds is $1 + \min_{d_i<0} \lfloor u_i/(-d_i) \rfloor$ when there is a negative coordinate. This follows by applying the same guard to $q + u + kd$ at the start of round k. With $d \geq 0$ the sequence can repeat indefinitely on an unbounded stock space. If $d = 0$, the coordinatewise peak across one round also gives a finite capacity.

MODEL E.01

Catalytic transport and installation

A transport component persists as a catalyst until the final installation. Stock coordinates are ordered b, f, n, p.

Original event sequence

Event	Input	Output	Before	After
transport	bf	bp	$(1,1,0,0)$	$(1,0,0,1)$
fabricate	bp	bn	$(1,0,0,1)$	$(1,0,1,0)$
install	bn	b	$(1,0,1,0)$	$(1,0,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,0,0)$ and net change $(0,-1,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. b at event 1, f at event 1.

A coupled event

Insert feed before the first original event, with input 1 and output f. The new minimum is $(1,0,0,0)$, the net change is $(0,0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	f	$(1,0,0,0)$	$(1,1,0,0)$
transport	bf	bp	$(1,1,0,0)$	$(1,0,0,1)$
fabricate	bp	bn	$(1,0,0,1)$	$(1,0,1,0)$
install	bn	b	$(1,0,1,0)$	$(1,0,0,0)$

MODEL E.02

Two catalyst units in one reaction

The first event requires two active copies of the same catalyst. Stock coordinates are ordered f, m, p, q.

Original event sequence

Event	Input	Output	Before	After
combine	mmf	mmp	$(1, 2, 0, 0)$	$(0, 2, 1, 0)$
refine	p	q	$(0, 2, 1, 0)$	$(0, 2, 0, 1)$
export	q	1	$(0, 2, 0, 1)$	$(0, 2, 0, 0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1, 2, 0, 0)$ and net change $(-1, 0, 0, 0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 1.

A coupled event

Insert feed before the first original event, with input 1 and output f. The new minimum is $(0, 2, 0, 0)$, the net change is $(0, 0, 0, 0)$, and the peak stock vector over one minimal round is $(1, 2, 1, 1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	f	$(0, 2, 0, 0)$	$(1, 2, 0, 0)$
combine	mmf	mmp	$(1, 2, 0, 0)$	$(0, 2, 1, 0)$
refine	p	q	$(0, 2, 1, 0)$	$(0, 2, 0, 1)$
export	q	1	$(0, 2, 0, 1)$	$(0, 2, 0, 0)$

MODEL E.03

A borrowed helper returned to the host

The preparation event keeps the replicator. The helper is present before assembly and remains afterwards. Stock coordinates are ordered f, m, p, v, z.

Original event sequence

Event	Input	Output	Before	After
prepare	fv	pv	$(1,1,0,1,0)$	$(0,1,1,1,0)$
assemble	mpv	mvz	$(0,1,1,1,0)$	$(0,1,0,1,1)$
release	z	1	$(0,1,0,1,1)$	$(0,1,0,1,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,0,1,0)$ and net change $(-1,0,0,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 2, v at event 1.

A coupled event

Insert feed after the first original event, with input 1 and output f. The new minimum is $(1,1,0,1,0)$, the net change is $(0,0,0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
prepare	fv	pv	$(1,1,0,1,0)$	$(0,1,1,1,0)$
feed	1	f	$(0,1,1,1,0)$	$(1,1,1,1,0)$
assemble	mpv	mvz	$(1,1,1,1,0)$	$(1,1,0,1,1)$
release	z	1	$(1,1,0,1,1)$	$(1,1,0,1,0)$

MODEL E.04

Use of a finite delivered part

The original round consumes one part without replacing it. Stock coordinates are ordered d, r, z.

Original event sequence

Event	Input	Output	Before	After
fit	dz	r	$(1,0,1)$	$(0,1,0)$
retire	r	d	$(0,1,0)$	$(1,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,0,1)$ and net change $(0,0,-1)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. d at event 1, z at event 1.

A coupled event

Insert deliver before the first original event, with input 1 and output z. The new minimum is $(1,0,0)$, the net change is $(0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1)$. With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
deliver	1	z	$(1,0,0)$	$(1,0,1)$
fit	dz	r	$(1,0,1)$	$(0,1,0)$
retire	r	d	$(0,1,0)$	$(1,0,0)$

MODEL E.05

Late delivery and initial inventory

The same delivery is scheduled at the end of the round. The first use must draw from stock. Stock coordinates are ordered d, r, z.

Original event sequence

Event	Input	Output	Before	After
fit	dz	r	$(1,0,1)$	$(0,1,0)$
retire	r	d	$(0,1,0)$	$(1,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,0,1)$ and net change $(0,0,-1)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. d at event 1, z at event 1.

A coupled event

Insert deliver after 2 original events, with input 1 and output z. The new minimum is $(1,0,1)$, the net change is $(0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
fit	dz	r	$(1,0,1)$	$(0,1,0)$
retire	r	d	$(0,1,0)$	$(1,0,0)$
deliver	1	z	$(1,0,0)$	$(1,0,1)$

MODEL E.06

Regeneration with two current units

Fabrication requires two copies of the same current machine as well as one boundary component. Stock coordinates are ordered b, f, m, n.

Original event sequence

Event	Input	Output	Before	After
fabricate	$mmbf$	$mmbn$	$(1,1,2,0)$	$(1,0,2,1)$
install	$mmbn$	mmb	$(1,0,2,1)$	$(1,0,2,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,2,0)$ and net change $(0,-1,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. b at event 1, f at event 1, m at event 1.

A coupled event

Insert feed before the first original event, with input 1 and output f. The new minimum is $(1,0,2,0)$, the net change is $(0,0,0,0)$, and the peak stock vector over one minimal round is $(1,1,2,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	f	$(1,0,2,0)$	$(1,1,2,0)$
fabricate	$mmbf$	$mmbn$	$(1,1,2,0)$	$(1,0,2,1)$
install	$mmbn$	mmb	$(1,0,2,1)$	$(1,0,2,0)$

MODEL E.07

One round that grows a reserve

The copied catalyst is retained rather than retired. Stock coordinates are ordered f, m.

Original event sequence

Event	Input	Output	Before	After
copy	mf	mm	$(1,1)$	$(0,2)$
sort	mm	mm	$(0,2)$	$(0,2)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1)$ and net change $(-1,1)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 1.

A coupled event

Insert feed before the first original event, with input 1 and output f. The new minimum is $(0,1)$, the net change is $(0,1)$, and the peak stock vector over one minimal round is $(1,2)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

All coupled net changes are nonnegative and at least one is positive. Indefinite unsaturated repetition increases that stock without bound.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	f	$(0,1)$	$(1,1)$
copy	mf	mm	$(1,1)$	$(0,2)$
sort	mm	mm	$(0,2)$	$(0,2)$

MODEL E.08

A release that keeps a bounded stock

An export event removes one of the two catalysts after copying. Stock coordinates are ordered f, m.

Original event sequence

Event	Input	Output	Before	After
copy	mf	mm	$(1,1)$	$(0,2)$
release	m	1	$(0,2)$	$(0,1)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1)$ and net change $(-1,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 1.

A coupled event

Insert feed before the first original event, with input 1 and output f. The new minimum is $(0,1)$, the net change is $(0,0)$, and the peak stock vector over one minimal round is $(1,2)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	f	$(0,1)$	$(1,1)$
copy	mf	mm	$(1,1)$	$(0,2)$
release	m	1	$(0,2)$	$(0,1)$

MODEL E.09

A substrate produced by a companion

A delivery placed after uptake leaves a separate initial feedstock guard. Stock coordinates are ordered a, b, f, p, q.

Original event sequence

Event	Input	Output	Before	After
uptake	af	ap	$(1,1,1,0,0)$	$(1,1,0,1,0)$
convert	bp	bq	$(1,1,0,1,0)$	$(1,1,0,0,1)$
finish	abq	ab	$(1,1,0,0,1)$	$(1,1,0,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,1,0,0)$ and net change $(0,0,-1,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. a at event 1, b at event 2, f at event 1.

A coupled event

Insert deliver after the first original event, with input 1 and output f. The new minimum is $(1,1,1,0,0)$, the net change is $(0,0,0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
uptake	af	ap	$(1,1,1,0,0)$	$(1,1,0,1,0)$
deliver	1	f	$(1,1,0,1,0)$	$(1,1,1,1,0)$
convert	bp	bq	$(1,1,1,1,0)$	$(1,1,1,0,1)$
finish	abq	ab	$(1,1,1,0,1)$	$(1,1,1,0,0)$

MODEL E.10

Preprocessing a secondary input

A persistent feed role and a consumable secondary substrate appear in separate coordinates. Stock coordinates are ordered f, m, p, t, w.

Original event sequence

Event	Input	Output	Before	After
preprocess	fw	fp	$(1,1,0,0,1)$	$(1,1,1,0,0)$
combine	mp	mt	$(1,1,1,0,0)$	$(1,1,0,1,0)$
reset	t	1	$(1,1,0,1,0)$	$(1,1,0,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,0,0,1)$ and net change $(0,0,0,0,-1)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 2, w at event 1.

A coupled event

Insert resupply before the first original event, with input 1 and output w. The new minimum is $(1,1,0,0,0)$, the net change is $(0,0,0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
resupply	1	w	$(1,1,0,0,0)$	$(1,1,0,0,1)$
preprocess	fw	fp	$(1,1,0,0,1)$	$(1,1,1,0,0)$
combine	mp	mt	$(1,1,1,0,0)$	$(1,1,0,1,0)$
reset	t	1	$(1,1,0,1,0)$	$(1,1,0,0,0)$

MODEL E.11

Controlled retirement after redundancy

The middle phase requires a temporarily redundant pair of machines. Stock coordinates are ordered f, m, w, z.

Original event sequence

Event	Input	Output	Before	After
duplicate	mf	mm	$(1,1,1,0)$	$(0,2,1,0)$
fabricate	mmw	mmz	$(0,2,1,0)$	$(0,2,0,1)$
replace	mmz	m	$(0,2,0,1)$	$(0,1,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,1,0)$ and net change $(-1,0,-1,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 1, w at event 2.

A coupled event

Insert feed before the first original event, with input 1 and output fw. The new minimum is $(0,1,0,0)$, the net change is $(0,0,0,0)$, and the peak stock vector over one minimal round is $(1,2,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	fw	$(0,1,0,0)$	$(1,1,1,0)$
duplicate	mf	mm	$(1,1,1,0)$	$(0,2,1,0)$
fabricate	mmw	mmz	$(0,2,1,0)$	$(0,2,0,1)$
replace	mmz	m	$(0,2,0,1)$	$(0,1,0,0)$

MODEL E.12

Transfer followed by recycling

The moving token circulates through three forms. An extra independent feed accumulates when imported. Stock coordinates are ordered c, f, x, y, z.

Original event sequence

Event	Input	Output	Before	After
transfer	cfx	cfy	$(1,1,1,0,0)$	$(1,1,0,1,0)$
consume	y	z	$(1,1,0,1,0)$	$(1,1,0,0,1)$
recover	z	x	$(1,1,0,0,1)$	$(1,1,1,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,1,0,0)$ and net change $(0,0,0,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. c at event 1, f at event 1, x at event 1.

A coupled event

Insert import after 2 original events, with input 1 and output f. The new minimum is $(1,1,1,0,0)$, the net change is $(0,1,0,0,0)$, and the peak stock vector over one minimal round is $(1,2,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is ∞ before the insertion and ∞ after it.

All coupled net changes are nonnegative and at least one is positive. Indefinite unsaturated repetition increases that stock without bound.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
transfer	cfx	cfy	$(1,1,1,0,0)$	$(1,1,0,1,0)$
consume	y	z	$(1,1,0,1,0)$	$(1,1,0,0,1)$
import	1	f	$(1,1,0,0,1)$	$(1,2,0,0,1)$
recover	z	x	$(1,2,0,0,1)$	$(1,2,1,0,0)$

MODEL E.13

A conversion loop with one leak

The split uses two copies, one from priming and one from stock or the coupled event. Stock coordinates are ordered m, p, q, v.

Original event sequence

Event	Input	Output	Before	After
prime	mv	mp	$(1,1,0,1)$	$(1,2,0,0)$
split	pp	q	$(1,2,0,0)$	$(1,0,1,0)$
restore	q	v	$(1,0,1,0)$	$(1,0,0,1)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,0,1)$ and net change $(0,-1,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. m at event 1, p at event 2, v at event 1.

A coupled event

Insert import after the first original event, with input 1 and output p. The new minimum is $(1,0,0,1)$, the net change is $(0,0,0,0)$, and the peak stock vector over one minimal round is $(1,2,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
prime	mv	mp	$(1,0,0,1)$	$(1,1,0,0)$
import	1	p	$(1,1,0,0)$	$(1,2,0,0)$
split	pp	q	$(1,2,0,0)$	$(1,0,1,0)$
restore	q	v	$(1,0,1,0)$	$(1,0,0,1)$

MODEL E.14

Cross-fed repair channels

Two retained machines use each other's intermediate to close the same round. Stock coordinates are ordered f, m, n, p, q.

Original event sequence

Event	Input	Output	Before	After
first	mf	mp	$(1,1,1,0,0)$	$(0,1,1,1,0)$
second	np	nq	$(0,1,1,1,0)$	$(0,1,1,0,1)$
renew	mnq	mn	$(0,1,1,0,1)$	$(0,1,1,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,1,1,0,0)$ and net change $(-1,0,0,0,0)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. f at event 1, m at event 1, n at event 2.

A coupled event

Insert feed before the first original event, with input 1 and output f. The new minimum is $(0,1,1,0,0)$, the net change is $(0,0,0,0,0)$, and the peak stock vector over one minimal round is $(1,1,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and ∞ after it.

The coupled round returns to its minimum initial stock. The peak vector therefore suffices as a fixed capacity for indefinite repetition.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
feed	1	f	$(0,1,1,0,0)$	$(1,1,1,0,0)$
first	mf	mp	$(1,1,1,0,0)$	$(0,1,1,1,0)$
second	np	nq	$(0,1,1,1,0)$	$(0,1,1,0,1)$
renew	mnq	mn	$(0,1,1,0,1)$	$(0,1,1,0,0)$

MODEL E.15

An installed part consumed by a test

A factory transforms one raw part into a replacement. The original round and the coupled round draw different stocks. Stock coordinates are ordered m, r, w, z.

Original event sequence

Event	Input	Output	Before	After
install	mz	mr	$(1,0,0,1)$	$(1,1,0,0)$
test	mr	m	$(1,1,0,0)$	$(1,0,0,0)$
inspect	m	m	$(1,0,0,0)$	$(1,0,0,0)$

A repeated letter denotes multiplicity. 1 denotes the empty input or output.

Minimal-stock certificate

The original sequence has minimum initial stock $(1,0,0,1)$ and net change $(0,0,0,-1)$. Starting at the minimum, the table records every intermediate stock.

Reducing one positive coordinate of that minimum breaks a guard, as follows. m at event 1, z at event 1.

A coupled event

Insert factory before the first original event, with input w and output z. The new minimum is $(1,0,1,0)$, the net change is $(0,0,-1,0)$, and the peak stock vector over one minimal round is $(1,1,1,1)$.

With two extra units above the appropriate minimum in every coordinate, the number of complete rounds is 3 before the insertion and 3 after it.

At least one stock coordinate loses units every coupled round. A fixed finite reserve in that coordinate bounds the repetition count.

Coupled sequence from its own minimum

Event	Input	Output	Before	After
factory	w	z	$(1,0,1,0)$	$(1,0,0,1)$
install	mz	mr	$(1,0,0,1)$	$(1,1,0,0)$
test	mr	m	$(1,1,0,0)$	$(1,0,0,0)$
inspect	m	m	$(1,0,0,0)$	$(1,0,0,0)$

CHAPTER 42

Local closure on finite context spaces

The final family calculates the demand comparison in the sheaf semantics of a finite poset. Every role is a constant label with an open region of demand and supply. Each transition has an open activation region. The upward closure of a displayed set of generators is the union of their principal opens.

The pointwise table gives the available data at each context. The closure locus is the largest open contained in the points where the comparison passes. A point survives this logical test exactly when all of its refinements pass. Viability of a finite graph is calculated pointwise by Theorem 14.4, and its image for state zero is an open. The joint locus is their intersection.

The tables therefore compare pointwise success, contextual truth, and eligible viability in a common presentation. The last column records a minimal generating antichain for each coupled open.

MODEL F.01

Two branches with different providers

One branch has its own repair, and the other gains a host repair. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 1$ on $\uparrow\{a\}$.

The host adds $0 \to 2$ on $\uparrow\{b\}$, $2 \to 2$ on $\uparrow\{b\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$
a	$\{m, r\}$	$\{m, r\}$	$\varnothing$	$\varnothing$	$\{0, 1\}$	$\{0, 1\}$
b	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a\}$	$\{a, b\}$	$\{a, b\}$
Viable start at state zero	$\{a\}$	$\{a, b\}$	$\{a, b\}$
Joint locus	$\{a\}$	$\{a, b\}$	$\{a, b\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
a	$0 \mid 1 \to 1$
b	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.02

A demand activated at the deepest refinement

A late demand makes all earlier contexts sensitive to a failure at the top. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $a \to b$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 1$ on $\uparrow\{a\}$.

The host adds $0 \to 0$ on $\uparrow\{b\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\{0,1\}$	$\{0,1\}$
b	$\{m,r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\{0,1\}$	$\{0,1\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\varnothing$	$\{a,b,o\}$	$\{o\}$
Viable start at state zero	$\{a,b\}$	$\{a,b\}$	$\{a\}$
Joint locus	$\varnothing$	$\{a,b\}$	$\{a\}$

In the endogenous catalog, $\{a,o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
a	$0 \mid 1 \to 1$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.03

A join of two partial contexts

The top context has both observational modes. Viability initially needs the whole path. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to t$, $b \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 2$ on $\uparrow\{b\}$, $2 \to 2$ on $\uparrow\{t\}$.

The host adds $1 \to 1$ on $\uparrow\{a\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{0,1\}$
b	$\{r\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$	$\varnothing$
t	$\{m,r\}$	$\{m,r\}$	$\{r\}$	$\varnothing$	$\{0,1,2\}$	$\{0,1,2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a,t\}$	$\{a,b,o,t\}$	$\{o\}$
Viable start at state zero	$\{t\}$	$\{a,t\}$	$\{a\}$
Joint locus	$\{t\}$	$\{a,t\}$	$\{a\}$

In the endogenous catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
a	$0 \mid 1 \to 1$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.04

A local loop outside the closure open

Idle operating loops exist at all contexts. One branch demands an unavailable completed repair. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $o \to c$. The base transition catalog is $0 \to 0$ on $\uparrow\{o\}$. The host adds only organizational supply in this model. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\{0\}$	$\{0\}$
a	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\{0\}$	$\{0\}$
b	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{r\}$	$\{0\}$	$\{0\}$
c	$\{m\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\{0\}$	$\{0\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a, c\}$	$\{a, c\}$	$\{a, c\}$
Viable start at state zero	$\{a, b, c, o\}$	$\{a, b, c, o\}$	$\{o\}$
Joint locus	$\{a, c\}$	$\{a, c\}$	$\{a, c\}$

In the endogenous catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

In the coupled catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
o	$0 \to 0$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.05

Host supply at a lower context

The host activates its contribution on the whole site and supports a separate loop. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to t$, $b \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 1$ on $\uparrow\{a\}$.

The host adds $0 \to 2$ on $\uparrow\{o\}$, $2 \to 2$ on $\uparrow\{o\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$
a	$\{m, r\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\{0, 1\}$	$\{0, 1, 2\}$
b	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$
t	$\{m, r\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\{0, 1\}$	$\{0, 1, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a, t\}$	$\{a, b, o, t\}$	$\{o\}$
Viable start at state zero	$\{a, t\}$	$\{a, b, o, t\}$	$\{o\}$
Joint locus	$\{a, t\}$	$\{a, b, o, t\}$	$\{o\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
o	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.06

Separate maximal contexts

Two chains have distinct activated requirements and two paths through the safe graph. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $a \to c$, $o \to b$, $b \to d$. The base transition catalog is $0 \to 1$ on $\uparrow\{a, c\}$, $1 \to 1$ on $\uparrow\{c\}$, $0 \to 2$ on $\uparrow\{b\}$, $2 \to 2$ on $\uparrow\{d\}$.
The host adds $1 \to 1$ on $\uparrow\{a\}$, $2 \to 2$ on $\uparrow\{b\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{m\}$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$	$\{0, 1\}$
b	$\{r\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$
c	$\{m\}$	$\{m\}$	$\{m\}$	$\varnothing$	$\{0, 1\}$	$\{0, 1\}$
d	$\{r\}$	$\{r\}$	$\{r\}$	$\varnothing$	$\{0, 2\}$	$\{0, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{c, d\}$	$\{a, b, c, d, o\}$	$\{o\}$
Viable start at state zero	$\{c, d\}$	$\{a, b, c, d\}$	$\{a, b\}$
Joint locus	$\{c, d\}$	$\{a, b, c, d\}$	$\{a, b\}$

In the endogenous catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.
The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
a	$0 \mid 1 \to 1$
b	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.07

An interface requiring a shared refinement

A recurrent cycle uses edges available on different principal opens. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to t$, $b \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 2$ on $\uparrow\{b\}$, $2 \to 0$ on $\uparrow\{t\}$.

The host adds only organizational supply in this model. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
b	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
t	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\{0, 1, 2\}$	$\{0, 1, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\varnothing$	$\{a, b, o, t\}$	$\{o\}$
Viable start at state zero	$\{t\}$	$\{t\}$	$\{t\}$
Joint locus	$\varnothing$	$\{t\}$	$\{t\}$

In the endogenous catalog, $\{a, b, o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
t	$0 \to 1 \to 2 \to 0$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.08

A dormant repair across a disconnected site

Two context components support different producers and no common refinement. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $a \to b$, $c \to d$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 1$ on $\uparrow\{b\}$.

The host adds $0 \to 2$ on $\uparrow\{c\}$, $2 \to 2$ on $\uparrow\{d\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
a	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$
b	$\{m, r\}$	$\{m, r\}$	$\varnothing$	$\varnothing$	$\{0, 1\}$	$\{0, 1\}$
c	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$
d	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{b\}$	$\{b, d\}$	$\{b, d\}$
Viable start at state zero	$\{b\}$	$\{b, d\}$	$\{b, d\}$
Joint locus	$\{b\}$	$\{b, d\}$	$\{b, d\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
b	$0 \mid 1 \to 1$
d	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.09

Contextual availability with a missing mode

Adding the host covers the machine demand on the right branch. The completed repair is still restricted. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to c$, $b \to c$, $b \to d$. The base transition catalog is $0 \to 1$ on $\uparrow\{o\}$, $1 \to 1$ on $\uparrow\{c\}$, $2 \to 2$ on $\uparrow\{d\}$.

The host adds $0 \to 2$ on $\uparrow\{b\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{r\}$	$\varnothing$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$
b	$\{m\}$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$
c	$\{m, r\}$	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{0, 1\}$	$\{0, 1\}$
d	$\{m\}$	$\varnothing$	$\{m\}$	$\varnothing$	$\{2\}$	$\{0, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{c\}$	$\{b, c, d\}$	$\{b\}$
Viable start at state zero	$\{c\}$	$\{c, d\}$	$\{c, d\}$
Joint locus	$\{c\}$	$\{c, d\}$	$\{c, d\}$

In the endogenous catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

In the coupled catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
c	$0 \mid 1 \to 1$
d	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.10

Demand on a strict subregion

Every demand is met at its active context. The safety graph gains an extra local loop. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $a \to b$, $a \to c$. The base transition catalog is $0 \to 1$ on $\uparrow\{o\}$, $1 \to 1$ on $\uparrow\{b\}$.

The host adds $1 \to 1$ on $\uparrow\{c\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
b	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\{0,1\}$	$\{0,1\}$
c	$\{r\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\{0,1\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a, b, c, o\}$	$\{a, b, c, o\}$	$\{o\}$
Viable start at state zero	$\{b\}$	$\{b, c\}$	$\{b, c\}$
Joint locus	$\{b\}$	$\{b, c\}$	$\{b, c\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
b	$0 \mid 1 \to 1$
c	$0 \mid 1 \to 1$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.11

Refinement closes a cycle

The endogenous round closes on one maximal branch. The host closes it on the other. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $a \to b$, $b \to c$, $b \to d$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 2$ on $\uparrow\{b\}$, $2 \to 0$ on $\uparrow\{c\}$.

The host adds $2 \to 0$ on $\uparrow\{d\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\varnothing$
b	$\{m, r\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\varnothing$
c	$\{m, r\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\{0, 1, 2\}$	$\{0, 1, 2\}$
d	$\{m, r\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 1, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{b, c, d\}$	$\{a, b, c, d, o\}$	$\{o\}$
Viable start at state zero	$\{c\}$	$\{c, d\}$	$\{c, d\}$
Joint locus	$\{c\}$	$\{c, d\}$	$\{c, d\}$

In the endogenous catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
c	$0 \to 1 \to 2 \to 0$
d	$0 \to 1 \to 2 \to 0$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.12

An everywhere active initial role

The initial machine has a wider demand than endogenous supply. The host expands the supply on one branch. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to t$, $b \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 1$ on $\uparrow\{t\}$.

The host adds $0 \to 0$ on $\uparrow\{b\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m\}$	$\varnothing$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$
a	$\{m, r\}$	$\{r\}$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$
b	$\{m\}$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$	$\{0\}$
t	$\{m, r\}$	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{0, 1\}$	$\{0, 1\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{t\}$	$\{b, t\}$	$\{b\}$
Viable start at state zero	$\{t\}$	$\{b, t\}$	$\{b\}$
Joint locus	$\{t\}$	$\{b, t\}$	$\{b\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
b	$0 \to 0$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.13

Two independent active cycles

The repair witness depends on which branch is observed. State-zero viability also uses branch-specific paths. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $o \to c$, $a \to t$, $b \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 0$ on $\uparrow\{a\}$, $2 \to 2$ on $\uparrow\{c\}$.
The host adds $0 \to 2$ on $\uparrow\{b, c\}$, $2 \to 2$ on $\uparrow\{b\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m, r\}$	$\{m\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$
a	$\{m, r\}$	$\{m, r\}$	$\varnothing$	$\varnothing$	$\{0, 1\}$	$\{0, 1\}$
b	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$
c	$\{m, r\}$	$\{m\}$	$\{r\}$	$\varnothing$	$\{2\}$	$\{0, 2\}$
t	$\{m, r\}$	$\{m, r\}$	$\{r\}$	$\varnothing$	$\{0, 1\}$	$\{0, 1, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a, t\}$	$\{a, b, c, t\}$	$\{a, b, c\}$
Viable start at state zero	$\{a, t\}$	$\{a, b, c, t\}$	$\{a, b, c\}$
Joint locus	$\{a, t\}$	$\{a, b, c, t\}$	$\{a, b, c\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
a	$0 \to 1 \to 0$
b	$0 \mid 2 \to 2$
c	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.14

An observational root with no local demand

The root passes its immediate demand test. Refinement exposes both demands and their witnesses. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to c$, $b \to c$, $c \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{c\}$, $1 \to 2$ on $\uparrow\{t\}$, $2 \to 2$ on $\uparrow\{t\}$.

The host adds $1 \to 1$ on $\uparrow\{c\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
a	$\{m\}$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$
b	$\{r\}$	$\varnothing$	$\{r\}$	$\varnothing$	$\varnothing$	$\varnothing$
c	$\{m, r\}$	$\{m\}$	$\{m, r\}$	$\varnothing$	$\varnothing$	$\{0, 1\}$
t	$\{m, r\}$	$\{m, r\}$	$\{m, r\}$	$\varnothing$	$\{0, 1, 2\}$	$\{0, 1, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{t\}$	$\{a, b, c, o, t\}$	$\{o\}$
Viable start at state zero	$\{t\}$	$\{c, t\}$	$\{c\}$
Joint locus	$\{t\}$	$\{c, t\}$	$\{c\}$

In the endogenous catalog, $\{o\}$ passes the pointwise demand comparison and is excluded from the closure open by a failing refinement.

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
c	$0 \mid 1 \to 1$

Each displayed lasso transports to every refinement of its generating context.

MODEL F.15

A fork below a common top

Two longer branches meet at the top. The topology records where their distinct repair paths extend. Roles m, r are a machine and a completed repair. The graph has safe states $0, 1, 2$.

Refinement and transition data

The refinement arrows are $o \to a$, $o \to b$, $a \to c$, $b \to s$, $c \to t$, $s \to t$. The base transition catalog is $0 \to 1$ on $\uparrow\{a\}$, $1 \to 2$ on $\uparrow\{c\}$, $2 \to 2$ on $\uparrow\{t\}$.
The host adds $0 \to 2$ on $\uparrow\{b\}$, $2 \to 2$ on $\uparrow\{s\}$. Each upward closure is taken in the refinement order.

Pointwise data and finite kernels

Context	Demand	Endogenous	Host	Missing after host	K	K_H
o	$\{m\}$	$\varnothing$	$\varnothing$	$\{m\}$	$\varnothing$	$\varnothing$
a	$\{m\}$	$\{m\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
b	$\{m, r\}$	$\varnothing$	$\{m\}$	$\{r\}$	$\varnothing$	$\varnothing$
c	$\{m, r\}$	$\{m, r\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
s	$\{m, r\}$	$\{m\}$	$\{m, r\}$	$\varnothing$	$\varnothing$	$\{0, 2\}$
t	$\{m, r\}$	$\{m, r\}$	$\{m, r\}$	$\varnothing$	$\{0, 1, 2\}$	$\{0, 1, 2\}$

Truth loci

Judgment	Endogenous	Host-coupled	Coupled generators
Organizational closure	$\{a, c, t\}$	$\{a, c, s, t\}$	$\{a, s\}$
Viable start at state zero	$\{t\}$	$\{s, t\}$	$\{s\}$
Joint locus	$\{t\}$	$\{s, t\}$	$\{s\}$

The last column lists the minimal contexts generating each coupled open. At those contexts the graph or rule catalog supplies a local certificate for every refinement in the open.

Local witnesses of coupled viability

Generating context	A coupled safe lasso from zero
s	$0 \mid 2 \to 2$

Each displayed lasso transports to every refinement of its generating context.

Bibliography

[1] *An* (∞)*-Topos-Theoretic Separation of Organisms, Viruses, and Computers*. Essay I in *Organisms, viruses, and computers*. Web reading edition. Read Essay I.

[2] *Operadic Autopoiesis in an* (∞)*-Topos*. Essay II in *Organisms, viruses, and computers*. Web reading edition. Read Essay II.

[3] *Thermodynamically Sustained Autopoiesis in an* (∞)*-Topos*. Essay III in *Organisms, viruses, and computers*. Web reading edition. Read Essay III.

[4] Jacob Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170, Princeton University Press, 2009. arXiv:math/0608040.

[5] Jacob Lurie. *Higher Algebra*. September 2017 version. Author's edition.

[6] Louis Martini and Sebastian Wolf. *Internal Higher Topos Theory*. 2023. arXiv:2303.06437.

[7] Denis-Charles Cisinski and Ieke Moerdijk. *Dendroidal sets as models for homotopy operads*. Journal of Topology 4 (2011), 257–299. arXiv:0902.1954.

[8] Gijs Heuts, Vladimir Hinich, and Ieke Moerdijk. *On the equivalence between Lurie's model and the dendroidal model for infinity-operads*. Advances in Mathematics 302 (2016), 869–1043. arXiv:1305.3658.

[9] John C. Baez and Kenny Courser. *Structured cospans*. Theory and Applications of Categories 35 (2020), 1771–1822. arXiv:1911.04630.

[10] John C. Baez and Blake S. Pollard. *A compositional framework for reaction networks*. Reviews in Mathematical Physics 29 (2017), 1750028. arXiv:1704.02051.

[11] John C. Baez, Brendan Fong, and Blake S. Pollard. *A compositional framework for Markov processes*. Journal of Mathematical Physics 57 (2016), 033301. arXiv:1508.06448.

[12] Alfred Tarski. *A lattice-theoretical fixpoint theorem and its applications*. Pacific Journal of Mathematics 5 (1955), 285–309. doi:10.2140/pjm.1955.5.285.

[13] F. William Lawvere. *Metric spaces, generalized logic and closed categories*. Rendiconti del Seminario Matematico e Fisico di Milano 43 (1973), 135–166. Open reprint.

[14] Dexter Kozen. *Results on the propositional* μ*-calculus*. Theoretical Computer Science 27 (1983), 333–354. doi:10.1016/0304-3975(82)90125-6.

[15] Matthew Hennessy and Robin Milner. *Algebraic laws for nondeterminism and concurrency*. Journal of the ACM 32 (1985), 137–161. Bibliographic record.

[16] Jean-Pierre Aubin. *A survey of viability theory*. SIAM Journal on Control and Optimization 28 (1990), 749–788. doi:10.1137/0328044.

[17] Patrick Saint-Pierre. *Approximation of the viability kernel*. Applied Mathematics and Optimization 29 (1994), 187–209. doi:10.1007/BF01204182.

[18] Francisco G. Varela, Humberto R. Maturana, and Ricardo Uribe. *Autopoiesis: the organization of living systems, its characterization and a model*. BioSystems 5 (1974), 187–196. PubMed record.

[19] Maël Montévil and Matteo Mossio. *Biological organisation as closure of constraints*. Journal of Theoretical Biology 372 (2015), 179–191. PubMed record.

[20] Matteo Mossio, Cristian Saborido, and Alvaro Moreno. *An organizational account of biological functions*. British Journal for the Philosophy of Science 60 (2009), 813–841. doi:10.1093/bjps/axp036.

[21] Robert Rosen. *Life Itself: A Comprehensive Inquiry into the Nature, Origin, and Fabrication of Life*. Columbia University Press, 1991. Publisher record.

[22] María Luz Cárdenas, Juan-Carlos Letelier, Claudio Gutierrez, Athel Cornish-Bowden, and Jorge Soto-Andrade. *Closure to efficient causation, computability and artificial life*. Journal of Theoretical Biology 263 (2010), 79–92. PubMed record.

[23] Tibor Gánti. *The Principles of Life*. Edited by Eörs Szathmáry and James Griesemer. Oxford University Press, 2003. Edition record.

[24] Wim Hordijk and Mike Steel. *Detecting autocatalytic, self-sustaining sets in chemical reaction systems*. Journal of Theoretical Biology 227 (2004), 451–461. PubMed record.

[25] Patrick Forterre. *Defining life: the virus viewpoint*. Origins of Life and Evolution of Biospheres 40 (2010), 151–160. Open article.

[26] Peter D. Nagy and Judit Pogany. *The dependence of viral RNA replication on co-opted host factors*. Nature Reviews Microbiology 10 (2012), 137–149. PubMed record.

[27] Derek Walsh and Ian Mohr. *Viral subversion of the host protein synthesis machinery*. Nature Reviews Microbiology 9 (2011), 860–875. Open article.

[28] Riccardo Rao and Massimiliano Esposito. *Nonequilibrium thermodynamics of chemical reaction networks: wisdom from stochastic thermodynamics*. Physical Review X 6 (2016), 041064. arXiv:1602.07257.

[29] Udo Seifert. *Stochastic thermodynamics, fluctuation theorems and molecular machines*. Reports on Progress in Physics 75 (2012), 126001. PubMed record.

[30] Jordan M. Horowitz, Kevin Zhou, and Jeremy L. England. *Minimum energetic cost to maintain a target nonequilibrium state*. Physical Review E 95 (2017), 042102. arXiv:1703.04722.

[31] Rodrigo Ramírez-Tagle. *Formalizing autopoiesis: toward a categorical-thermodynamic calculus of closure*. BioSystems (2026), 105918. PubMed record.

[32] Richard M. Karp. *Reducibility among combinatorial problems*. In *Complexity of Computer Computations*, pp. 85–103. Plenum Press, 1972. doi:10.1007/978-1-4684-2001-2_9.