A Higher-Categorical Theory of Honest Distributed Computer Networks

Operational conformance, descent, higher witnesses, and progress

Anonymous

10 November 2026

Abstract

Honesty is conformance of every performed local transition to a public specification, relative to its actual input and the declared environment. This monograph separates that property from the existence of a compatible global execution, the choice of a comparison between executions, and the eventual delivery of an observable result. An observation site carries a sheaf of possible executions and a subobject of honest ones. Composition over shared interfaces is homotopy pullback, so a matching witness remains part of the composed state. The theory derives an extension criterion over fixed boundaries, a local replacement theorem for open systems, an obstruction calculus for abelian and nonabelian matching, and a prefix-to-run theorem with a distinct fairness hypothesis. It connects these constructions to crash-stop replication, consistent cuts, knowledge, quorum agreement, convergence, commit, capacity, and timed constraints. A companion atlas gives exact finite instances, with independently checkable witnesses for the computations. **Subjects.** Higher topos theory, distributed computing, directed concurrency, homotopy limits, fault models.

Contents

1 The subject of the theory 2

2 Operational records and honesty 5

3 The observation site 9

4 Completion and compositionality 13

5 Higher matching and obstruction theory 16

6 Refinement and directed semantics 21

7 Causality, schedules, and fair runs 24

8 Observation, knowledge, and internal logic 30

9 Agreement, quorums, and limits of honesty 33

10 From local rules to service contracts 37

11 A calculus of observable conformance 41

12 Exact methods for the case atlas 44

13 Atlas of causal schedules 48

14 Atlas of quorum availability 77

15 Atlas of timed boundary constraints 106

16 Atlas of abelian matching obstructions 133

17 Atlas of finite matching groupoids 157

18 Atlas of closed and open relay diagrams 192

19 Atlas of distributed knowledge 218

20 Atlas of finite progress certificates 237

21 Atlas of capacity and higher schedule cells 249

22 Using and extending the theory 267

1 The subject of the theory

A sender records that it transmitted a message, and a receiver records that it has heard nothing. Each record can be faithful. Whether the pair describes one execution depends on the channel, the cut at which each record was taken, and the rule for relating the two cuts. Given a reliable asynchronous channel and a cut containing the send, the message can be in transit. Given a round whose specification requires delivery before the receiver's logged checkpoint, the records can be incompatible. The same local statements therefore have different joint meanings in different observation contexts.

There is a further distinction even in an execution with no faults. A receiver can rename local handles on restart while preserving every payload and every permitted dependency. Comparing a pre-restart record with a post-restart one can require a choice of isomorphism of handles. Two choices can yield different identifications on a second link. If a global verifier records only classes of handles, those choices have already disappeared before the verifier gets to check a cycle. A higher-categorical description keeps the witnesses until their effect on the question has been determined.

1.1 Four questions for one network

Fix a collection of computers, code, interfaces, observation maps, and environment rules. Four questions concern different mathematical objects.

1. *Conformance.* Is each performed step allowed by the transition rule with the inputs available at that step?

2. *Compatibility.* Is there a single execution or coordinated record realizing all the observations, and what is the space of such realizations?

3. *Refinement.* Which witnesses make a change of representation correct in every allowed context, including after wiring and hiding?

4. *Progress.* For which inputs and schedulers do admissible finite records extend to runs that deliver the required outcome?

Conformance has an invariant interpretation as a subobject. Compatibility is a homotopy-limit question with boundary conditions. Refinement is relative to observable maps. Progress quantifies over

admitted runs and depends on the environment's guarantee about time, scheduling, or delivery. A network can satisfy any one of these properties in settings where another fails. An object describing all four should therefore carry the data for each.

Definition 1.1 (Honest computer network)**.** An honest computer network is a specified collection of processes and channels together with admissible environment behavior. In an honest run every performed process transition belongs to the process's transition relation, applied to its actual local state and input. Every performed channel or environment step belongs to its own declared relation. A crash, omission, timeout, or recovery is honest when the selected model permits that event and its declared effects.

The term has a technical meaning throughout. For example, a channel that may omit messages still obeys its relation. An application that requires delivery uses a different assumption about the channel. Admissibility of a crash event says nothing about whether a process eventually restarts. The specification can require an honest step to attach evidence, but production of that evidence and truth of an arbitrary statement about the environment remain different assertions.

This definition is neutral about implementation technology. An operating system event, a remote procedure call, a store-and-forward relay, and a finite-state replicated object can all be expressed using local transitions. The effect of the transition is meaningful only with respect to an interface. A process may have an internal counter that is private to its scheduler. The same counter can become part of an observable epoch identifier after a change to the protocol.

1.2 An architecture of mathematical objects

Let C be a small category of observations and J a topology of jointly exhaustive observations. Let $\mathcal{E} = \mathrm{Sh}_{\mathcal{S}}(C, J)$. Over this base the raw record object X and its honest part H fit into a monomorphism $H \hookrightarrow X$. An interface is a map $H \to B$. A second map $W \to H$ can specify a space of evidence for each honest record. The image of $W \to H$ says which records admit some evidence. The fiber records all choices of evidence and their identifications. A time filtration is a further diagram $H_0 \leftarrow H_1 \leftarrow \cdots$, or a corresponding diagram in spaces over a fixed input. An admissible scheduler selects a subspace of, or a collection of representatives in, the resulting space of infinite runs.

Two systems with observations in common can be composed without first taking their set of connected components. Their common observations determine a diagram, and the composed record is a limit of the diagram. A point specifies the component records and matching paths, with higher coherences when the diagram requires them. Given a fixed external boundary, the space of completions is the corresponding fiber. A contractible fiber means that there is a completion unique up to a contractible choice of all comparisons. An empty fiber is a compatibility failure. A disconnected fiber distinguishes multiple solutions. Higher homotopy distinguishes different ways those solutions can be compared.

The directed category of executions plays a separate role. Its arrows append steps or compare histories in a declared direction. The homotopy type of a record at a fixed boundary does not encode whether a send can be undone. A comparison that merely renames identifiers can be invert-

ible without any corresponding reversal of causal time. The model therefore separates variation of observation contexts, forward execution, and equivalence of representations.

1.3 Statements developed here

The following results organize the theory. Each specifies the input that makes the conclusion valid.

1. Downward collections of cells give an observation site on which every diagram of cell data extends by a limit formula. The extension satisfies Čech descent. For fixed boundary conditions the same construction computes a fiber of global completions.
2. A conformance predicate descends whenever it is stable under restriction and membership is local for the specified covers. Existence of evidence is a weaker datum than a coherent family of evidence. The distinction persists even for finite patch systems.
3. A change of representation over the same interface preserves all gluing results exactly when it is an equivalence in the corresponding slice. Local contractibility of relative fibers yields such an equivalence, and boundary probes detect it.
4. For a finite graph of groupoid-valued patch data, gauge fixing on a spanning forest reduces the remaining constraints to holonomy. For abelian higher coherence specified by a cocycle, the space of lifts is empty exactly when the obstruction class is nonzero. When lifts exist their components and higher automorphisms are computed in lower cohomological degrees.
5. A genuine finite event poset gives a contractible cubical space of partially executed events, and its linear extensions are related by swaps of adjacent incomparable events. Capacity restrictions can delete independent squares and change these statements about the restricted system.
6. Finite branching gives existence of compatible infinite prefixes under explicit nonemptiness hypotheses. Fairness and termination require separate conditions, which can be expressed using obligations and ranking functions. A finite safety certificate alone supplies no fairness guarantee.

The combination is useful because most distributed specifications move between all these modes. A storage replica can be locally honest, furnish an unavailable quorum, carry a nontrivial recovery identifier, and expose a reader to an indefinitely delayed message in the same run. A proof that treats every issue as agreement in a single set loses information about which hypothesis failed.

1.4 Relation to established theories

Lamport's happened-before relation gives the ordering of observed events in the asynchronous model [9]. Chandy and Lamport explain how channel state enters a consistent snapshot [10]. The asynchronous computability theorem of Herlihy and Shavit uses chromatic protocol complexes and simplicial decision maps [12]. Those complexes track indistinguishability of process views. Our observation-site model can use the same finite views as input, but its completion fibers retain explicit interface paths when the views carry symmetries. An equivalence between these models would require a construction matching both their observations and their morphisms.

Higher-dimensional automata and directed topology model the relationship be-

tween concurrency and higher-dimensional cells [13]. The construction in Section 7 starts from a finite poset of events and calculates one explicit cubical model. We specify when a higher cell has an operational interpretation. The sheaf construction in Section 3 instead concerns descent across observing regions. A directed cube of execution and a Čech simplex of compatibility have different meanings even when they have the same dimension.

The ambient higher-category theory is standard [1, 2, 3]. Cellular sheaves on finite spaces already encode relations between local data and restrictions [7, 8]. The extra input here is the honest operational semantics, the choice of what an interface remembers, and a treatment of progress over the same architecture. Classical impossibility results remain theorems about their specified models. We use the deterministic asynchronous consensus obstruction [15] to identify a limit of the passage from local conformance to global liveness.

1.5 Reading conventions and scope

Spaces are homotopy types. An equality in a space is a path, with higher comparisons understood. An equality in a specified set is ordinary equality. Limits in categories of spaces, sheaves of spaces, and slices are their homotopy-invariant limits. In the finite examples with discrete data these coincide with the corresponding ordinary limits. We write $\mathrm{fib}_b(f)$ for the homotopy fiber over a point b, including its path to b. A point of a sheaf over a general stage U is a map from its representing sheaf, so all fiber criteria can be stated without assuming enough ordinary points.

We use a universe of small spaces and small indexing categories whenever size matters. Statements using choices of lifts or bases are external unless stated internally. Finite operational examples specify their alphabet and predicates explicitly. The proofs are mathematical arguments. Exact enumerations illustrate the formulas and supply certificates for finite claims. The examples do not assume that a physical network, a probability distribution, or a machine-checked formalization has been measured.

1.6 A notation map

Symbol	Interpretation
I	category of cells, with arrows from faces to cells
C, J	observation category and coverage
W_U	subterminal presheaf of cells visible on U
X, H	raw records and the honest subobject
B, W	interface and evidence objects
$\mathrm{fib}_b(H \to B)$	completions with boundary b
P	finite poset of executed events
$J(P)$	poset of downsets of P
T_n	n-stage histories with truncations
Γ	graph or global sections, as indicated
BG	one-object groupoid of a group G
$K(A, n)$	Eilenberg–Mac Lane space

The dual use of Γ follows standard notation. A graph Γ has vertices and edges. A global-section expression $\Gamma(U, H)$ always names the observation stage and the object. The symbol $J(P)$ is unrelated to the Grothendieck topology J.

2 Operational records and honesty

2.1 Local machines and actual inputs

Let V be a set of processes and E a directed multigraph of channels, including a channel identifier when parallel channels have distinct semantics. At process v let S_v be the local state set, I_v the set of input events, and O_v the set of finite output batches. A process rule is a relation

$$R_v \subseteq S_v \times I_v \times S_v \times O_v.$$

An idle step can be represented by a distinguished input, and its availability is a separate part of the rule. A nondeterministic transition has several possible outcomes. The rule's membership condition checks the chosen outcome. To model probabilistic selection one equips the possible transitions with a specified kernel and separately checks the distributional assumption.

For each channel e, let Q_e be its buffer state space. A reliable FIFO channel with payload alphabet M_e has state $Q_e = M_e^*$, an append transition on send, and a pop transition on receive when the buffer is nonempty. A multiset buffer gives unordered delivery. An omission channel also has a delete transition authorized by the environment rule. Duplication can be represented by an explicit copy transition, or by a persistent delivery token with a rule that permits several receives. For a retrying protocol the request identifier is part of the payload or the channel metadata.

The ambient state

$$S = \prod_{v \in V} S_v \times \prod_{e \in E} Q_e \times Z$$

includes any state Z of time, inputs, crash flags, or delivery permissions. One step acts on the local coordinates and the specified channel coordinates. In the atomic formulation a send both updates the sender and enqueues the output. A receive both pops an authorized message and supplies that value to the receiving transition. A different granularity with enqueue and local update separate requires an intermediate state for pending output. The formulations can be compared by a refinement only if the intermediate state is hidden at the public interface and its added paths preserve the declared failure semantics.

Definition 2.1 (Execution record)**.** A finite raw execution record is a finite sequence of labeled steps with recorded pre-states, post-states, and inputs, indexed by a specified initial state. It is honest if the adjacent state records agree and every step is authorized at its actual state. The environment supplies input events that do not come from channels.

Input authenticity in this sense is a condition in the model of the trace. It can be checked by retaining the actual source occurrence of an input. In a model with anonymous interchangeable messages one may instead retain only the payload and multiplicity. These representations induce different maps to observations. When two send occurrences carry the same payload, choosing which occurrence enabled a receive may be invisible in one representation and part of the history in another.

Proposition 2.2 (Exact local check)**.** *For a linear record with shared state variables, conformance holds if and only if all initialization, transition, frame, and adjacency constraints hold. Here a frame constraint says that each coordinate outside the step's write set remains unchanged.*

Proof. An execution of the product transition system restricts to one authorized transition with the required frame at each time index. The initial state and the state passed from one index to the next

give the initialization and adjacency constraints. Conversely, the initialization constraint chooses an allowed starting state. Given the state before step i, the adjacency condition identifies it with the pre-state in the local constraint at i. The transition rule updates the coordinates in its write set and the frame fixes all others. Induction constructs the specified sequence of global states. Channel consistency is part of the transition rule at the steps that change or inspect the channel. □

If a record omits coordinates rather than exposing the full product state, these constraints define a space of completions. A shared coordinate at an overlap must occur as one variable, or the overlap must specify how its occurrences are compared. Without this requirement the proof establishes only that each isolated record is consistent.

2.2 Message evidence and matching

A send carries an identifier m, source s, destination d, and payload a. A receive mentions m when its interface records provenance. A constrained event structure can express validity by requiring that every receive have a predecessor send with the same key, channel, and payload. With FIFO ordering, a second condition requires matching receives to respect the order of prior sends on the same channel. An exactly-once delivery rule forbids reuse of a send occurrence by distinct receives. A reliable finite prefix can still contain a sent message with no receive.

If message identifiers are generated locally, use a namespace with origin, incarnation, and sequence number. This is a mathematical naming assumption. We do not require any particular cryptographic primitive. The identifier may designate an equivalence class of handles rather than an immutable string. Then matching a send with a receive includes a comparison in the groupoid of such handles.

Example 2.3 (Two indistinguishable payloads)**.** Suppose a sender emits two messages with equal payload a and an unordered channel delivers both. The receiver records two positions, each labeled a. With distinguished send occurrences and no other constraint there are two matchings, the identity and the transposition. If the sender and receiver both expose FIFO order there is one matching. If sender occurrence names may be permuted and this symmetry acts on the matching data, the two matchings form one free orbit. The corresponding action groupoid is contractible. A quotient on payloads alone has the same number of connected components in the last two models, but it gives no account of which comparisons were allowed.

Example 2.4 (An overlap loses correlation)**.** A source chooses one bit x and sends it to both listeners. The first listener's observation is the sent value at its port, and the second listener's observation is the sent value at its port. Each projected interface permits both values. The joint boundary admits only $(0, 0)$ and $(1, 1)$. Replacing the source by two independent interfaces with the same marginal values admits $(0, 1)$. A gluing proof must preserve the joint boundary map rather than separately preserving its two component images.

2.3 Crash-stop and recovery

A crash-stop model equips each process with an active flag. An environment transition clears the flag. Process transitions require it to be set. The model can allow messages already in transit from a crashed process to arrive. Crash recovery adds an epoch and a specified projection from volatile state to durable state. A restart

reinitializes volatile state from the durable record and increments the epoch if epochs are part of the specification. Outstanding messages need explicit rules for incarnation matching, duplicate execution, and receiver interpretation.

Proposition 2.5 (Persistence across epochs)**.** *Suppose a restart map reads only durable state. If an invariant P depends only on durable state, holds initially, and every active transition preserves it, then every run preserves P provided crash and restart leave durable state unchanged.*

Proof. Induct on the record length. At an active step use the preservation hypothesis. At crash or restart the durable state is identical, so the predicate has the same value. At a channel or other environment step use the frame condition on durable coordinates. □

An invariant of both durable and volatile state requires a separate restart proof. For example, a volatile promise not to accept ballots below a number can disappear at recovery, even though every individual promise before and after recovery was honestly made by the process. A protocol that requires such promises to persist can store them durably or distinguish configurations and epochs so that an obsolete promise is never used in an active decision.

Example 2.6 (Three possible crash cuts)**.** A process receives an increment request, stores the new total, and sends a reply. If a crash occurs before the store, a retry must still apply the increment for the request to succeed. If it occurs after the store but before the reply, applying the increment again changes the total twice. If it occurs after the reply, a delayed duplicate may still arrive. The three cases differ only by where the crash intersects the causal chain. An atomic durable update storing both the total and the request identifier makes the latter two cases return the stored result, provided retention of identifiers covers the retry interval.

2.4 Sequential witnesses and projections

Let T be the rooted category of finite histories ordered by prefix. An event appends a step $h \to h'$. A map $p : T \to O$ to an observation category can forget internal actions or keep an abstract label. When O is a poset of external prefixes, an internal action maps to an identity. External steps map to monotone extensions. The fiber over an observation is allowed to have many histories. A public observer can know whether an input was received without knowing which private buffer allocation handled it.

For a directed map p, two useful lifts should be distinguished. An object lift chooses a history for a given observation. A path lift extends a *given* history along a specified external step. The first condition can hold without the second. Given an implementation that commits to a hidden bit b before the first visible output, both one-step outputs can appear in the image. A history in which $b = 0$ was already chosen cannot lift the external choice 1 if the implementation forces its next output to equal b. This distinction is relevant to online refinement, availability, and scheduler coupling.

Definition 2.7 (Prefix simulation)**.** For labeled transition systems T, T' with an agreed interface, a forward simulation is a relation on states preserving initial states and public observations such that for every related pair (s, t) and each step from s, there is a matching finite path from t ending at a related state. Whether a path may be empty and which labels are hidden

are part of the specification.

By induction, a forward simulation lifts every finite execution from a chosen related starting state to a matching path. The converse from inclusion of finite traces fails when the choice of a related implementation state would need future information. For infinite runs, a simulation that permits empty matching paths needs a divergence or fairness condition to avoid matching infinitely many steps by a finite path. Such a condition is additional to the relation above.

2.5 Run assumptions as separate parameters

We distinguish three parameter lists.

1. The local transition rules specify what a step does.
2. The environment selects initial conditions and which external or fault events may happen.
3. The scheduler selects admissible orders and, for infinite runs, delivery and progress obligations.

Their combinations define the mathematical quantifiers. For an external input b let R_b be the honest runs and let $F_b \subseteq R_b$ be the fair ones. Feasibility asks whether R_b is inhabited. A fair outcome may ask whether some point of F_b has the property. A guaranteed outcome asks whether every point of F_b has it, with a separate condition ensuring that F_b is nonempty. A verification can also quantify over all inputs in a designated set. These three levels of quantifiers are required to compare protocols meaningfully.

A useful interface can expose which fairness clauses apply. A relay proof may guarantee progress for each request if every enqueued request eventually reaches a live destination. Composing it with a queue whose workers crash and never restart fails that assumption even if every remaining action is honest. The failure is detected by comparing the explicit environment guarantees to the assumptions of the relay.

3 The observation site

3.1 From records to a site

An observation site records what a verifier can restrict and which families exhaust an observation. Choose the following cells for a single message exchange. There are state observations before and after the send, observations before and after the receive, and an interface observation for the message. The transition cells include the state and message observations as faces. A patch containing a transition must contain its faces. Its faces need not expose all private variables of that transition. Overlaps can refer to the same message cell without recording one process's private state in the other process's patch.

A model can instead use patches of a topology on agents, cells of a run complex, or stages in a stack of configurations. The cover condition is meaningful only after the overlap data have been specified. A collection of one-processor records is generally insufficient if some channel constraint is invisible in every record. In the example one either puts message interface cells in the relevant overlaps or includes an explicit channel patch. Otherwise the cover models a verifier of a strictly weaker record.

Definition 3.1 (Finite patch site)**.** Let I be a finite poset, with $i \le j$ meaning that i is a face of j. Let $C(I)$ be its poset of lower subsets. A family $U_a \subseteq U$ covers U if $U = \bigcup_a U_a$. Write I_U for the full subposet on U.

Intersections of lower subsets are lower. For every inclusion $V \subseteq U$, intersections $V \cap U_a$ cover V. Composing covering families preserves the union condition. Thus the coverage defines a Grothendieck topology. The empty family covers the empty patch. This forces a sheaf to assign a terminal object to the empty patch. That terminal object represents the single empty record.

3.2 The cellular section theorem

A contravariant cell diagram $D : I^{\mathrm{op}} \to \mathcal{S}$ gives a type of possible data for each cell and a coherent restriction map to each face. Define a presheaf by

$$H_D(U) = \lim_{I_U^{\mathrm{op}}} D|_{I_U^{\mathrm{op}}}. \tag{1}$$

An elementary point of the limit in a set-valued example is a compatible assignment $x_i \in D(i)$ to all cells in U. In general it also contains paths, homotopies relating the paths along composable face maps, and the rest of their coherent compatibility.

Theorem 3.2 (Cellular descent)**.** *For every $D : I^{\mathrm{op}} \to \mathcal{S}$, the presheaf H_D is a sheaf on $C(I)$. Restriction to the principal lower sets $\downarrow i$ and the construction $D \mapsto H_D$ define inverse equivalences between $\mathrm{Sh}_{\mathcal{S}}(C(I))$ and $\mathrm{PSh}_{\mathcal{S}}(I)$. In particular,*

$$H_D(U \cup V) \simeq H_D(U) \underset{H_D(U \cap V)}{\times} H_D(V). \tag{2}$$

Proof. In $\mathrm{PSh}_{\mathcal{S}}(I)$ let W_U be the subterminal presheaf with value a point on U and the empty space elsewhere. The category of elements of W_U is I_U. Mapping out of the colimit of representables that presents W_U gives

$$\mathrm{Map}(W_U, D) \simeq \lim_{I_U^{\mathrm{op}}} D.$$

For a cover (U_a), let $S_\bullet$ be the Čech nerve of $\coprod_a W_{U_a} \to W_U$. Colimits and limits of presheaves of spaces are computed objectwise. At a cell outside U the augmentation is the identity of the empty space. At a cell in U the nerve is the Čech nerve of a nonempty discrete set mapping to a point. This is the nerve of the groupoid with one arrow between each ordered pair of its objects, so its realization is contractible. The augmentation $\mathrm{colim}_{\Delta^{\mathrm{op}}} S_\bullet \to W_U$ is an equivalence. Mapping out gives Čech descent, including (2).

For the equivalence, $I_{\downarrow i}^{\mathrm{op}}$ has initial object i, so $H_D(\downarrow i) \simeq D(i)$. To see that these values determine every sheaf K, use induction on $|U|$. If U is empty, both values are terminal. If U is principal, there is nothing to prove. Otherwise remove one maximal cell i. The two proper lower sets $U \setminus \{i\}$ and $\downarrow i$ cover U; their intersection is proper. The restriction map from K to the limit extension of its principal values is an equivalence on these three smaller sets by induction, hence on U by descent. The constructions and the equivalences are natural in K and D. □

Corollary 3.3 (Constant cell labels)**.** *For a constant diagram with value L, one has $H_D(U) \simeq \mathrm{Map}(|NI_U|, L)$, where N is the nerve.*

Proof. The limit of the constant diagram is the mapping space from the colimit of the constant point diagram to L. That colimit is the classifying space of I_U^{op}, canonically equivalent to the classifying space of I_U by reversal. □

The word constant refers here to the *cell* diagram. For example, if U represents a circle and $L = BG$, its value on the whole circle is $\mathrm{Map}(S^1, BG)$. The presheaf that simply assigns BG to every nonempty

patch would generally have a different value. The change is the precise source of the loop information in the examples below.

3.3 Arrows of a site and arrows of a protocol

An inclusion of observation patches $U \subset V$ produces a restriction $H(V) \to H(U)$. A directed run extension $h \to h'$ may append an action observed entirely within a fixed patch. Thus the site and the time category need not coincide. One can choose a site with objects (U, n), where n is an observation horizon and U a visible patch. Restriction along U forgets spatial information, and restriction along n truncates time. It is then a separate assertion that the two restriction operations commute. They do in the obvious coordinate model if both are specified by deleting entries from a single ordered record. They need not commute with a derived observation such as *last acknowledged write* if the observation scheme also drops supporting messages.

The use of an ∞-topos extends the finite-site treatment. A sheaf of spaces on a Grothendieck site is a presheaf satisfying descent for covering sieves, equivalently for the associated Čech covers when the topology is presented in that form [1, Chapters 5–6]. In a general site a hypercover has matching conditions in successive degrees. If the intended principle is descent along all hypercovers, work with hypercomplete sheaves or justify the appropriate hyperdescent directly. The finite union proof in Theorem 3.2 only uses its displayed Čech augmentation.

3.4 Conformance as a subobject

Let X be a sheaf of raw records. For a stage U, suppose P_U is a predicate on $X(U)$ invariant under paths. If $x \in X(U)$ then $P_U(x)$ is either true or false externally. Internally the same condition is a proposition in the stage. A specification of conformance is *stable* if every restriction of a record satisfying it satisfies the restricted condition. It is *local* if satisfaction on a covering family implies satisfaction on U.

Theorem 3.4 (Local conformance descends)**.** *If P is stable and local, the full subspace $H(U) \subseteq X(U)$ on the components satisfying P_U defines a sheaf and a monomorphism $H \hookrightarrow X$. Conversely, a subobject of X gives such a predicate.*

Proof. Stability gives restriction maps. Fullness means that if a record lies in $H(U)$, all paths in $X(U)$ with endpoints that record and another record in $H(U)$ lie in $H(U)$. For a cover, apply descent to a compatible family regarded as data in X. Its glued record satisfies P by locality. The same descent of paths holds because the inclusions are full. Hence H satisfies descent, and the inclusions are monomorphisms since all their homotopy fibers are empty or contractible.

Conversely, membership means that the fiber of $H(U) \to X(U)$ at the record is inhabited. This map is a monomorphism of spaces because evaluation preserves limits. Restriction preserves membership. If each local fiber is inhabited, it is contractible. Their descent diagram has a contractible limit, since every object in the diagram is contractible. Applying descent to the fiber diagram shows that the global fiber is contractible. □

Example 3.5 (Shared-state conformance)**.** Let each cell of a finite site be a state coordinate, a process transition, or a channel operation. In a transition cell retain the state and interface coordinates needed by Proposition 2.2. Take a raw diagram with all finite tuples for each cell and the usual projections to faces. At transition cells

restrict to allowed tuples. At shared faces one can leave all raw values. The subdiagram defines an honest sheaf by Theorem 3.2; it is a subobject of the raw sheaf because limits preserve monomorphisms. The resulting global condition is exactly simultaneous satisfaction of all local constraints.

This construction can include bounds, for example an invariant that a local durable counter stays within a declared interval. A global capacity constraint such as the sum of two counters may require a cell that sees both. If no cover element sees it and its truth cannot be inferred from the overlap, it does not define a local property for that chosen cover. It can still be defined on the global records or expressed in a stronger site.

3.5 Evidence and the semantics of existence

Let $p : W \to H$ carry evidence. A path $h \simeq h'$ in $H(U)$ transports evidence from the fiber over h to the fiber over h'. Even when every honest record admits evidence, the space of evidence may have nontrivial automorphisms. Form the image factorization in $\mathcal{E}$

$$W \twoheadrightarrow \mathrm{im}(p) \hookrightarrow H.$$

The first arrow is an effective epimorphism. The image states that the evidence exists locally. A lift to W over a stage is a chosen global witness at that stage.

Example 3.6 (Locally valid identifiers)**.** Let a local record name an object with one of two labels. On a circular observation region let transport around the circle swap the labels. Every sufficiently small interval admits a label. The space of global labels is empty, since a global label would have to be fixed by the swap. The local truth that a label exists gives a global section of the image proposition. It does not supply a section of the two-sheeted local system.

The example already occurs with sets as fibers. For groupoid fibers there may be a nonempty set of global witness classes but several automorphisms of each. In an ∞-topos, propositional truncation of the fiber is the right operation when a verifier asks only whether witnesses locally exist. Taking global sections first answers whether there is a witness on the entire stage. The order of operations matters.

3.6 Locality and change of base

A geometric morphism has an inverse image functor preserving finite limits and colimits. Consequently it preserves terminal objects, pullbacks, monomorphisms, effective epimorphisms, and images. For the last assertion, apply the inverse image functor to the Čech nerve of the effective epimorphism and to its realization. This carries the factorization above to an effective epimorphism followed by a monomorphism, which is the image factorization up to equivalence.

This observation lets one transport a finite interface proof along an inverse image functor. It does not imply preservation of arbitrary limits such as a tower of infinite schedules. For example, an inverse image taking a stalk is a filtered colimit of sections over neighborhoods. Filtered colimits in spaces commute with finite limits. For an infinite family of conditions, each germ can hold on its own neighborhood, with no neighborhood on which all conditions hold at once. A change-of-base claim for fairness or infinite histories therefore needs a hypothesis beyond left exactness.

4 Completion and compositionality

4.1 Fixed boundaries

An open network record has a boundary map $f : H \to B$. The boundary may consist of input and output events, or of several sets of visible values and timestamps. If $b : U \to B$ is a stage-indexed observation, its completion object is

$$H_b = U \times_B H \longrightarrow U.$$

A section of $H_b \to U$ is a record together with a comparison between its boundary and b. This allows the boundary to carry paths, and therefore applies to a matching problem where observed identifiers may be renamed.

For two records $H_1 \to M \leftarrow H_2$ sharing an interface, their joint record is $H_1 \times_M H_2$. Over a fixed stage, its points can be represented as

$$(h_1, h_2, \gamma), \qquad \gamma : f_1(h_1) \simeq f_2(h_2).$$

When there are also external maps to B_1 and B_2, the resulting external boundary is a map to $B_1 \times B_2$. Fixing external boundary values commutes with this construction by associativity of pullbacks. The result holds as an equivalence of relative objects, with all projections.

Theorem 4.1 (Relative gluing)**.** *Let H, B be sheaves on a site and let $b \in B(U)$. For any Čech cover $U_\bullet \to U$ for which both have descent, the completion space satisfies*

$$\mathrm{fib}_b(H(U) \to B(U)) \simeq \mathrm{Tot}\, F_\bullet,$$
$$F_n = \mathrm{fib}_{b_n}(H(U_n) \to B(U_n)).$$

Here b_n is the restriction of b in degree n. When that degree is a family of overlaps, its space of sections means the product of sections over those overlaps.

Proof. The pullback $* \times_{B(U)} H(U)$ is the limit of a span. Replace each of its three vertices by the limit of its restriction diagram. The restrictions of the constant point diagram have limit a point. The two indexing limits commute in spaces, as can be checked by mapping a test space into either order of limits and comparing cones. The inner pullbacks are the displayed fibers. □

Descent compares a global space with a *compatible* family of local spaces. A product of the local spaces, or a list recording that each is inhabited, does not encode the family on the right. A point of the totalization includes higher compatibility. With one common discrete interface the compatibility is equality of port values. With one common groupoid interface it is a specified isomorphism.

4.2 Interfaces as spans

A two-sided system is a span $A \leftarrow H \to B$. Compose it with $B \leftarrow K \to C$ to obtain

$$A \longleftarrow H \times_B K \longrightarrow C.$$

The units are the identity spans. For three composable spans, the two parenthesizations represent the limit of the same zigzag diagram, with the same maps to the endpoints. The space of comparison equivalences preserving the limiting cones is contractible. The same observation for longer strings supplies coherent associativity. It gives a category of spans up to equivalence and the usual higher structure when comparisons are retained. For a full construction of higher spans with finite limits see [5].

There is a tensor product given by independent parallel placement. Its action on boundary objects and apex objects is cartesian product. The pullback universal

property gives the interchange equivalence

$$(H \times H') \times_{B \times B'} (K \times K') \simeq (H \times_B K) \times (H' \times_{B'} K').$$

This tensor treats the components as independent except for their declared interfaces. If they share a capacity register or a common input, that dependence should appear in the base of a pullback. Example 2.4 exhibits an observable difference between a product and a correlated placement.

Proposition 4.2 (Feedback with a declared match)**.** *Let $H \to A \times M \times B \times M$ carry two ports of type M. Constraining the ports to match gives*

$$H \times_{M \times M} M \longrightarrow A \times B,$$

where the map from M is the diagonal. Over fixed $h \in H$ the fiber is the path space between its two port values. If the output port represents a later step than the input port, this construction enforces equality of the specified values. To model causal feedback, include a delay or a state relation in place of the diagonal.

Proof. A point of the fiber is m with paths from both port values to m. The space of choices of m with a path from the first value is contractible. Eliminating that choice leaves a path between the two port values. Replacing the diagonal by a relation uses the same universal property with the relation's apex. □

4.3 A boundary test for exact replacement

For a fixed interface B, compare $X \to B$ and $Y \to B$ by a map $u : X \to Y$ in $\mathcal{E}_{/B}$. A context is a record $Z \to B$ with which the system can be joined. The comparison in context is $Z \times_B u$. The slice records both the map and its homotopy over B; comparison of apexes alone would lose that datum.

Theorem 4.3 (Context and boundary detection)**.** *The following are equivalent.*

1. *u is an equivalence over B.*
2. *$Z \times_B u$ is an equivalence for every $Z \to B$.*
3. *For every site stage U and every $b \in B(U)$, the induced map of completion spaces at b is an equivalence.*

In the third condition, for each (U, b) it suffices to compare the relative objects after restriction to a cover of U. Each comparison there is an equivalence of sheaves on the restricted site, or equivalently on all its test stages.

Proof. Equivalences are stable under pullback, giving the second condition, and taking $Z = B$ gives its converse. Evaluation of sheaves preserves limits, so the first condition implies the third. For the converse, fix U. A map of spaces over $B(U)$ is an equivalence when it is an equivalence on every homotopy fiber, for example by the classification of spaces over $B(U)$ as space-valued families indexed by $B(U)$. Thus $u(U)$ is an equivalence for every U. A natural transformation of presheaves that is objectwise an equivalence is an equivalence. For the last assertion, use Theorem 4.1 on both sides. Equivalence of completion objects over the cover pulls back to all intersections, and their totalizations agree. If the hypothesis is expressed by equivalences on all generalized fibers on the cover, this is the same equivalence of relative sheaves. Equivalently one can apply the local argument directly to the fibers of u over each test section of Y, using contractibility. □

Here local checking must apply with the quantifier over generalized points. For

example, one can require that for each $y \in Y(U)$, a cover of U make the relative fibers of u over y contractible on its Čech diagram. Then the same limit argument makes the fiber over y contractible. A test of one selected boundary value at each physical device is typically a smaller collection of tests.

Corollary 4.4 (Gluing replacement along a finite cover)**.** *Suppose a map of open records restricts to an equivalence over the same boundary on every intersection in a finite patch cover. The map on global completion spaces at every fixed boundary is an equivalence. The result is natural under wiring to any additional records on that boundary.*

Proof. Apply Theorem 4.1 to the cover, obtaining a levelwise equivalence between the two diagrams of fibers. Limits preserve equivalences. For wiring, pull the resulting relative equivalence back along the newly attached records. □

4.4 Forest reduction for open records

Consider a finite undirected tree whose vertices have record spaces X_v, whose edges have interface spaces B_e, and whose ends have specified maps $X_v \to B_e$. An assignment is a point in the limit of this diagram. Choose a root r. For every nonroot vertex v, let S_v be the record space for the subtree rooted at v and let $p_v : S_v \to B_{e_v}$ be its map to the edge toward its parent. Construct recursively

$$S_v \simeq X_v \times_{\prod_w B_{e_w}} \prod_w S_w, \tag{3}$$

where w ranges over the children of v. The maps $S_w \to B_{e_w}$ and the ports of X_v form the pullback.

Proposition 4.5 (Tree elimination)**.** *The root record in (3) is the limit of the entire tree diagram. If every p_v is an equivalence, the projection $S_r \to X_r$ is an equivalence. If all spaces are finite sets, the number of completions over x_v is the product of the child completion counts at the port values of x_v.*

Proof. A leaf subtree has no child constraints and record space X_v. A cone on the subtree at v consists of a cone on each child subtree, a point of X_v, and comparisons at the child interfaces. This is represented by the pullback in (3). Induction proves the first statement. If the child maps to the ports are equivalences, their product is an equivalence and so is its pullback. In finite sets, taking the fiber at x_v gives a product of finite child fibers. Counting gives the final formula. □

The finite formula sums over hidden labels only after multiplying child counts. It is valid with nonnegative weights in any commutative semiring, as distributivity permits elimination one variable at a time. Shared labels across two branches violate the independence needed for this recursion unless they are included in the separator data. The atlas gives examples with one closing edge where one must retain a pair of terminal labels.

4.5 A truncation counterexample

In spaces, choose the same base point on a circle twice. The homotopy pullback of $* \to S^1 \leftarrow *$ is $\Omega S^1 \simeq \mathbb{Z}$ as a discrete homotopy type. Each of the three spaces in the input diagram has just one component. Taking components first and then pulling back gives a singleton. Thus there is a difference between taking components of a composed network and composing component sets. Even when the output will be

a set, the composition can depend on the retained input paths.

A finite variant uses $* \to BG \leftarrow *$ for a finite group G. The pullback has $|G|$ distinct components, indexed by actual matching isomorphisms. The automorphism group at the unique object of BG becomes a set of choices in the completion fiber. The effect of a higher choice can therefore be detected by an ordinary count.

Proposition 4.6 (Truncated completions)**.** *If every map in a family $(f_i : X_i \to Y_i)$ indexed by one category is n-truncated, the induced map of limits is n-truncated. In particular, base change of an n-truncated boundary map is n-truncated.*

Proof. A fiber of a map of limits is a limit of the fibers at a coherent family of base points. The full subcategory of n-truncated spaces is closed under limits. For instance, this follows by induction from the characterization by the truncation of the diagonal, starting with equivalences. The base-change statement follows directly by identifying fibers. □

There is no parallel general assertion that a limit of connected spaces is connected. The circle example is already a limit of connected spaces with disconnected result. When using an existence result based on connectivity, it matters whether connectivity was established after all constraints were imposed.

4.6 A minimal invariant for closed observations

Let $W \subseteq H$ be a witnessed safety property expressed as a subobject. For a boundary b, say the network guarantees W at b if $W_b \to H_b$ is an equivalence. For meaningful existence one may additionally require H_b to be inhabited. A comparison u between networks can preserve this guarantee by pulling back W and testing the corresponding fiber maps. If u is an equivalence over the joint boundary, the guarantee and the completion space are unchanged in every context. If u only preserves the image of feasible boundary values, the guarantee need not be preserved: two records can realize the same external value while only one exhibits the safety property. The evidence retained in the safety predicate determines which kind of replacement the proof needs.

5 Higher matching and obstruction theory

5.1 The role of a matching path

When an interface is a set, its diagonal tests equality and there is at most one comparison. If it is a groupoid, the comparison can be one of several isomorphisms. A process can follow its prescribed local rules while leaving a choice of coordinate frame, a recovery identifier, or a permutation of unlabeled resources. A witness for composition says how those choices relate. The witness itself can have automorphisms.

The first calculations use the one-object groupoid BG of a group G. If G is finite it can be represented as a finite list of permitted relabelings with a multiplication table. It is the interface of *one* object with G ways to compare it with itself. There is no stochastic or adversarial assumption associated with this choice.

5.2 Graph gauge theorem

Choose an orientation of each edge of a finite graph Γ. Consider a model with one copy of BG at each vertex, the same interface on every edge, and identity restriction maps. Equivalently use the constant cell diagram of Corollary 3.3 on a graph subdivided if necessary to a simplicial complex.

An object of the graph matching groupoid is a family $g = (g_e) \in G^{E(\Gamma)}$. A morphism $g \to g'$ is a family $(h_v) \in G^{V(\Gamma)}$ with

$$g'_e = h_{t(e)} g_e h_{s(e)}^{-1}. \tag{4}$$

A path in the graph transports by multiplying the edge labels in order of function composition, inverting a label if the path traverses its edge backwards.

Theorem 5.1 (Forest gauge and residual conjugation)**.** *Let Γ have c components, and let $r_j = |E_j| - |V_j| + 1$ for component j. Its matching groupoid is equivalent to*

$$\prod_{j=1}^{c} (G^{r_j} /\!\!/ G),$$

where G acts by simultaneous conjugation. If one vertex in each component is framed, so its gauge coordinate must be the identity, the groupoid is equivalent to the discrete space $\prod_j G^{r_j}$.

Proof. Choose a spanning tree and root in each component. Starting with $h_{\mathrm{root}} = 1$, solve (4) recursively to make the tree edge labels identities. If the tree edge goes from parent to child, the child's gauge coordinate is the parent's times the inverse label. If it goes from child to parent, the child's coordinate is the parent's times the label. Thus every object is isomorphic to one in the full subgroupoid with tree labels identities. Between objects in that subgroupoid, (4) on tree edges says that the gauge is constant. The remaining edge labels transform by simultaneous conjugation. Framing requires that constant to be the identity. □

Each g_e on a nontree edge in the reduced picture is the holonomy of a based loop determined by that edge and its two tree paths. A different spanning tree changes the free generators of the fundamental group. A choice of root and tree gives a presentation of the matching groupoid, while the equivalence class of the groupoid is independent of those choices.

Example 5.2 (A cycle with three overlapping records)**.** Put a copy of BC_m on each member of a cyclic observation diagram, where $C_m = \mathbb{Z}/m\mathbb{Z}$. Orient all three edges around the cycle. An edge comparison has value $a_i \in C_m$. The invariant is $a_1 + a_2 + a_3$. Each invariant yields one connected component with automorphism group C_m. If there is also a filled comparison triangle, the composition must be the identity, so only the zero sum remains. If a comparison with specified nonzero sum was part of the external boundary, no filled record has that boundary.

5.3 Relations from two-dimensional patches

Let X be a connected CW complex with a finite graph as its 1-skeleton and with 2-cells attached along closed edge words. A constant BG target gives $\mathrm{Map}(X, BG)$. The graph picture adds a relation for each 2-cell: the holonomy along its attaching loop is the identity. Since BG is a 1-type, there are no further choices of filling a loop once the boundary is fixed and the loop is null. For a path-space comparison between loops with specified endpoints, this is the assertion that an equality between paths in a 1-type is a proposition.

Corollary 5.3 (Presentation of groupoid–valued lifts)**.** *For such a connected X,*

$$\mathrm{Map}(X, BG) \simeq \mathrm{Hom}(\pi_1(X, x), G) /\!\!/ G.$$

If the target object is framed at x, the based mapping space is the discrete set $\mathrm{Hom}(\pi_1(X, x), G)$.

Proof. Use Theorem 5.1 on the graph. Each 2-cell imposes its attaching word

on the remaining free generators. Higher cells attach along spheres of dimension at least two; the restriction problem into a 1-type has contractible fibers for these attachments. Thus the based lifts are homomorphisms from the presented group. Removing the frame retains the global conjugation action. □

The same statement for an arbitrary connected homotopy type uses its fundamental groupoid, via the left adjoint from spaces to 1-types. It is often simplest to compute the ordinary presentation and then reintroduce the conjugation action. Quotienting by conjugation before a boundary condition has been imposed is not generally permissible. For a two-port comparison one may have fixed frames at both ports, so the path between them carries relative information.

Proposition 5.4 (Centralizers and counting)**.** *For a finite G and a finitely generated group Λ, put $S = \mathrm{Hom}(\Lambda, G)$. The automorphism group at a representation ρ in the unframed matching groupoid is $C_G(\rho(\Lambda))$. Its component count is*

$$|\pi_0(S /\!\!/ G)| = \frac{1}{|G|} \sum_{g \in G} |\,\mathrm{Hom}(\Lambda, C_G(g))|. \tag{5}$$

Its groupoid cardinality, the sum of reciprocals of automorphism orders over components, is $|S|/|G|$.

Proof. A gauge fixes ρ exactly when it commutes with every value of ρ. Count pairs (g, ρ) with $g \cdot \rho = \rho$ first by g, giving the sum in (5). The contribution of one orbit is its size times its stabilizer size, which is $|G|$. The cardinality formula follows by dividing the size of each orbit by $|G|$. □

These two counts answer different questions. The first counts inequivalent ways to match. The second is additive under disjoint union and accounts for internal automorphisms. For instance, the groupoid BG has one component and groupoid cardinality $1/|G|$. The atlas reports both quantities and identifies the stabilizer sizes in the finite group examples.

5.4 From cocycles to boundary obstructions

Let A be abelian and $n \geq 1$. A $K(A, n)$-valued coherence can attach an A-valued choice to degree n. The failure to close in the next degree is an A-valued cocycle. In a finite CW presentation, choose an orientation for each relevant cell. Write $d : C^k(X; A) \to C^{k+1}(X; A)$ for the coboundary. For a prescribed $(n+1)$-cocycle c, the primary equation is

$$da = c, \qquad a \in C^n(X; A). \tag{6}$$

Two solutions can differ by a cocycle. Changes of $(n-1)$-cochain make an elementary comparison, and lower cochains provide its higher comparisons.

Theorem 5.5 (An abelian completion space)**.** *Let X be a CW complex, $n \geq 1$, and let $\kappa : X \to K(A, n+1)$ represent a class $[\kappa] \in H^{n+1}(X; A)$. Let L_κ be the space of nullhomotopies of κ to the constant zero map. Then L_κ is inhabited precisely when $[\kappa] = 0$. After choosing a nullhomotopy,*

$$L_\kappa \simeq \mathrm{Map}(X, K(A, n)).$$

Its components form a torsor under $H^n(X; A)$. For every chosen component and base point,

$$\pi_i(L_\kappa) \cong H^{n-i}(X; A) \quad (1 \leq i \leq n),$$

and the higher groups vanish.

Proof. The component set of $\mathrm{Map}(X, K(A, n+1))$ represents

$H^{n+1}(X;A)$. The space of paths from κ to zero is empty unless their components agree. Choosing one path identifies all such paths with the loop space at zero. The functor $\mathrm{Map}(X,-)$ preserves limits, so the loop space is $\mathrm{Map}(X,\Omega K(A,n+1))$. This yields the equivalence. Translation by loops gives the free and transitive action on components. The remaining homotopy groups are found by iterating loops and taking components, using $\Omega^i K(A,n) \simeq K(A,n-i)$ through $i=n$. Beyond $i=n$ the iterated loop space is contractible. □

Corollary 5.6 (Relative coefficients)**.** *If $Y \subseteq X$ is a CW subcomplex and a boundary nullhomotopy on Y is specified, the extension problem is controlled by the relative class in $H^{n+1}(X,Y;A)$ defined by the map and that boundary homotopy. When extensions exist, their component torsor and homotopy groups are the corresponding relative groups.*

Proof. The homotopy fiber at zero of the restriction $\mathrm{Map}(X,K(A,n+1)) \to \mathrm{Map}(Y,K(A,n+1))$ is the pointed mapping space from the cofiber X/Y. The given boundary path defines a point in this fiber. Nullhomotopies there are precisely extensions with a comparison to the specified boundary path. Apply the pointed form of the preceding proof and use $\widetilde{H}^*(X/Y;A) = H^*(X,Y;A)$, taking the cofiber to include the base point for $Y=\varnothing$. □

There is a useful sign check for relative problems. In a cochain model let $a_Y \in C^n(Y;A)$ satisfy $da_Y = c|_Y$, and extend it arbitrarily to $\widetilde{a} \in C^n(X;A)$. Then $c - d\widetilde{a}$ lies in $C^{n+1}(X,Y;A)$. Replacing $\widetilde{a}$ by another extension changes this cochain by a relative coboundary. Solving for a relative correction b in $db = c - d\widetilde{a}$ is equivalent to solving (6) with $a|_Y = a_Y$.

5.5 Three different obstruction mechanisms

Example 5.7 (Disk with pinned boundary)**.** Let $X=D^2$, $Y=S^1$, $A=\mathbb{Z}$, and $n=1$. Give the boundary a degree-m map to $K(\mathbb{Z},1)$. An extension across the disk would be a nullhomotopy. Equivalently the boundary class has image m under the connecting isomorphism

$$H^1(S^1;\mathbb{Z}) \longrightarrow H^2(D^2,S^1;\mathbb{Z}).$$

When $m=0$, the relative groups H^1 and H^0 vanish, so the completion space is contractible. The unrestricted disk does not have this obstruction.

Example 5.8 (A relation with torsion)**.** Attach a 2-cell to a circle by a map of degree $m>0$. The cochain map $C^1(X;A) \to C^2(X;A)$ is multiplication by m. Hence $H^1 = \ker(m : A \to A)$, $H^2 = A/mA$, and $H^0 = A$. With $A=\mathbb{Z}/\ell\mathbb{Z}$, $\ell>0$, the first two groups have order $\gcd(m,\ell)$. The primary equation $ma=c$ has a solution exactly when that divisor divides an integer representative of c. At $m=6,\ell=8$ an odd c cannot be completed, and any even c gives two component classes. In contrast a coefficient group of odd order coprime to 6 gives a unique component for every c.

Example 5.9 (Parallel face constraints)**.** Give two oriented 2-cells the same attaching word on a one-loop graph. Over an abelian group let the induced coboundary from one edge coordinate be $a \mapsto (a,a)$. Each constraint has a solution on its own. The pair (c_1,c_2) has a solution exactly when $c_1=c_2$. The obstruction is the difference of the two values. A verifier that checks only local solvability of the two faces

loses this difference. The honest local steps can still give rise to the incompatible proposed boundary.

5.6 Twisted coefficients on a loop

A local coefficient can change when transported around the observation cycle. Let $T : A \to A$ be an automorphism. Consider the bundle over S^1 with fiber $K(A, n)$ and monodromy induced by T. Its space of sections is the homotopy equalizer of id and T acting on $K(A, n)$. A point in it consists of a value at one cut and a path from that value to its image by transport. A model of $K(A, n)$ as a simplicial abelian group identifies this equalizer with the homotopy fiber at zero of $T - \mathrm{id}$.

Proposition 5.10 (Cycle transport)**.** *For the section space L above,*

$$\pi_n(L) \cong \ker(T - \mathrm{id}),$$
$$\pi_{n-1}(L) \cong \operatorname{coker}(T - \mathrm{id}),$$

and all other nonnegative homotopy groups vanish. At $n = 1$ the second formula describes the component group.

Proof. The fiber sequence for $T - \mathrm{id}$ gives its long exact homotopy sequence. The source and target have a single nonzero homotopy group in degree n, on which the map is $T - \mathrm{id}$. Exactness gives the kernel and cokernel. At degree zero the fiber is a simplicial abelian group, so the component set has the indicated group structure. □

Example 5.11 (Orientation reversal)**.** Take $A = \mathbb{Z}/8\mathbb{Z}$ and $T(a) = -a$. Then $T - \mathrm{id} = -2$. Its kernel is $\{0, 4\}$ and its image is the even subgroup. For $n = 2$, the lift space is connected and has two-element groups in degrees one and two. The trivial monodromy case instead yields eight-element groups in both degrees. All local fibers and all fiberwise transition types are the same in the two cases. The loop transport distinguishes the global witness types.

5.7 When obstruction classes apply

The cocycle theorem computes a particular lifting problem. For an arbitrary map of record spaces it is not enough to compute the cohomology of the physical communication graph. A construction must identify the relevant domain X, coefficient system, and relative lifting problem. For instance, a graph of exact payload equalities may give an obstruction to finding any assignment, but the appropriate calculation can be a system of finite relations with no preferred abelian structure.

For a nilpotent or otherwise controlled target one can sometimes proceed by a Postnikov tower. After a lift to one stage is chosen, the next stage imposes a further condition determined by its k-invariant and the induced local coefficients. Changing the earlier lift can change that condition. A contractible choice of intermediate comparisons is a stronger result than mere existence. In the general case, a convergent tower and transport of coefficients need to be justified rather than presumed from a finite set of computations.

5.8 Operational interpretation of an obstruction

A nonzero class in the prescribed relative problem says that the requested boundary and local comparisons have no simultaneous realization. It does not identify a particular processor as the source of error. An external observer may have combined snapshots from different epochs. The observer may have imposed a constant frame on a system whose interface has a twisted local system. The model may have omitted a channel operation required by the com-

putation. Each of these can produce an empty completion fiber while every actual step satisfies its local rule.

There is a more informative report than a single yes-or-no result. A certificate can list the boundary values, the comparisons allowed on each overlap, and a closed cycle or cocycle pairing that rules out the specified realization. A positive result can supply a completion and describe its alternatives by a torsor, orbit, or finite solution set. These certificates attach to the particular observation model that generated them.

6 Refinement and directed semantics

6.1 Families indexed by observed state

A map $p : X \to B$ of spaces can be read as a family $b \mapsto X_b$. A path in B gives an equivalence between the two endpoint fibers, and the composites agree coherently. Conversely, a family of spaces indexed by the ∞-groupoid B assembles to a map to B. This is the groupoidal form of straightening and unstraightening [1, Section 2.2]. It explains why the fibers in Theorem 4.3 determine equivalence over the base. Counting their connected components would classify a strictly coarser object.

When the index is a directed category T, a functor $T^{\mathrm{op}} \to \mathcal{S}$ gives transport from a later context to an earlier one. Its Grothendieck construction is a right fibration. A functor $T \to \mathcal{S}$ gives transport in the other direction and a left fibration. A directed extension with several possible successors is not, in general, a function between the state sets. It can be represented by a span of possible moves, or by a total category whose arrows record those moves. One needs an actual transport law to use the fibration description. For example, forgetting the last event of a prefix gives a definite truncation. Adding a next event requires a choice and may be unavailable.

6.2 An execution category with declared comparisons

Consider a category T of directed histories and a class W of arrows that are intended to be only representation changes. A functor $p : T \to O$ to a public observation category determines which of these changes are invisible. Require that $p(w)$ be an equivalence for every $w \in W$. Localization gives

$$T \longrightarrow T[W^{-1}] \xrightarrow{\bar{p}} O.$$

It is characterized by the property that functors out of $T[W^{-1}]$ correspond to functors out of T that send W to equivalences. This characterization retains coherent natural transformations. It is the reason to use ∞-categorical localization when there can be higher choices of comparison [4, 2].

For a poset of visible prefixes, the only equivalences are identities. The observation of a localized history cannot move backwards along a strict visible prefix extension. A path in the localized category may include inverses of invisible changes. The composite has the same public order as its directed visible part. This condition says nothing about which hidden observations have been erased, so the choice of O and W belongs to the specification.

Proposition 6.1 (Observation of a localized arrow)**.** *Let O be an ordinary poset regarded as an ∞-category. If $w \in W$ preserves p exactly, an arrow $x \to y$ in $T[W^{-1}]$ implies $p(x) \le p(y)$. If $p(x) < p(y)$, that arrow is not an equivalence.*

Proof. Since $p(W)$ consists of identities, localization provides $\bar{p}$. An arrow in a

poset exists only in the indicated direction. An equivalence is sent by any functor to an equivalence, and a strict inequality in a poset is not an equivalence. □

For instance, two buffer layouts can be connected by an invertible reallocation that preserves the sequence of sent and received messages. Sending an additional observable message still creates a strict extension. Erasing all arrows by passing to the groupoid completion would also invert that extension. Such a groupoid completion describes a different question about the undirected shape of the state space.

6.3 Fiber criteria and structural witnesses

Let $X \to Y \to B$ be a record transformation followed by observation. Suppose that for every generalized stage point $y : U \to Y$ the relative space of elaborations of y to X is contractible, locally on U and on its test stages. Then $X \to Y$ is an equivalence by descent on the elaboration fibers. This criterion is strictly stronger than the condition that each y has an elaboration. If a new record includes a handle with a free choice of label, the labels can form a contractible action groupoid when every relabeling is permitted. If the same handle has nontrivial stabilizer, the elaboration fiber is a classifying space and is not contractible.

Example 6.2 (Allocation with unused labels)**.** A model uses a fixed finite set L of m labels and allocates a distinguished object one label. Suppose all permutations of L are allowed and the unused labels remain part of the representation. The allocation groupoid is $L /\!\!/ S_m$. It is connected, with stabilizer S_{m-1}. For $m \geq 3$, forgetting allocation is therefore not an equivalence, despite the single component. If unused labels are first forgotten and comparisons remember only the bijection on the one used label, there is one arrow between any two choices. The resulting groupoid is contractible. Which refinement is valid depends on whether the context can observe the unused part.

Example 6.3 (Interleavings as elaborations)**.** Two actions operate on disjoint coordinate sets with no shared resource constraint. An implementation can use a sequential log containing either order. If those orders are considered distinct and the specification has only one undirected history, its elaboration space has two points. If one adds the comparison induced by the commuting square and its inverse, and imposes no other comparison, the two objects form the free connected groupoid on one edge. That groupoid is contractible. With three or more actions, multiple swap routes can produce loops. An equivalence claim must account for their higher coherences.

6.4 Replacement under hiding and composition

A boundary projection $B \to B'$ forgets some visible information. It induces a functor $\mathcal{E}_{/B} \to \mathcal{E}_{/B'}$ by postcomposition. Equivalences over B remain equivalences after hiding. This follows from functoriality and does not require hiding to preserve arbitrary limits. The order of hiding and wiring can still matter. To connect two systems on a specific identifier one must wire while that identifier remains exposed.

Theorem 6.4 (Stable elimination of refinements)**.** *Take a diagram of honest record objects and interfaces in an ∞-topos, with an external boundary cocone specified as maps from the record limit. A map of such diagrams, with the same external boundary, induces an equivalence on closed records in every added context if it is an equivalence*

at every vertex and interface. The conclusion continues to hold after hiding any subset of boundary coordinates. For a diagram presented by descent, it is sufficient to have the equivalences over every member of a cover as equivalences of relative sheaves.

Proof. The comparison of diagrams is an equivalence in the functor category. Taking the limit gives an equivalence and respects the specified external maps. A context is applied by pullback and hiding by composition. Both functors preserve equivalences. For descent, apply Theorem 4.1 and local detection over the interfaces before taking the diagram limit. □

A common use of the theorem is eliminating an internal representation that has no effect on the interface even after further wiring. One can certify an elaboration by a pointwise section and a coherent contraction of its fibers. Giving only a section yields a retraction, but there may be other components or loops. Conversely, one can show that a proposed elimination is invalid by a single boundary probe whose completion fiber changes.

6.5 Retaining failures when projecting

A failure model can itself be a boundary. Let F classify schedules of fault events at a chosen granularity, and let $H \to B \times F$ record both the external trace and the declared faults. Suppose an implementation introduces a different failure event representation $F' \to F$. The correct comparison over F' is with $H \times_F F'$, not with H after forgetting F. For example, an implementation may allow a recovery between local update and message enqueue while the atomic specification has neither observation nor fault cut there. A comparison over B can identify all successful outcomes and miss that difference.

Proposition 6.5 (Relative fault comparison)**.** *Suppose $X \to B \times F'$ maps to $H \times_F F' \to B \times F'$ and is an equivalence over this base. For each declared fault record f' and each external observation b, its space of completions agrees with that of H at $(b, j(f'))$, where $j : F' \to F$ is the fault map.*

Proof. Pull back the equivalence along the point $(b, f') \to B \times F'$. The target fiber is the specified fiber by iterated pullback. □

The proposition quantifies over faults already present in F'. It does not show that $F' \to F$ admits every fault record of F. If fault coverage is part of correctness, state it separately, for example as a surjectivity or lifting condition on the external fault family.

6.6 Coherent replacement versus forward simulation

A map of record objects that is an equivalence over all finite observations can be useful for safety. Progress can still depend on which record extensions are permitted and which runs are fair. If the map comes from a natural equivalence of prefix towers, it induces an equivalence of the limit spaces. To transport a universal fair property, the comparison must also identify the fair subspaces or at least send every fair run of the implementation to a fair run of the specification in the direction of the proof.

Proposition 6.6 (Transfer of a fair guarantee)**.** *Let $u : R \to R'$ be a map of admitted runs over their common external input. Suppose u maps every fair run of R to a fair run of R'. Let the outcome predicate on R be the pullback of the one on R'.*

If all fair runs of R' satisfy the predicate, then all fair runs of R satisfy it. If an existence guarantee is required for R, fair inhabitation of R is a separate hypothesis.

Proof. Apply the universal guarantee to the image of each fair run of R. Pullback of the predicate gives the conclusion at the original run. The proof does not construct an input run. □

If an implementation can perform a private step forever, a projection can map its finite prefixes to the same public prefix. Extending the projection to infinite runs may then require adjoining stuttering to the specification. Adjoining stuttering does not automatically make such a run fair. A refinement proof with termination can exclude unbounded stuttering by a well-founded rank that decreases on private steps as long as no visible progress occurs.

6.7 An interface for quantitative guarantees

Suppose finite record witnesses carry costs in a commutative ordered monoid M, with cost additive under serial composition and monotone under the relevant observation map. For an external pair (a, b), let $C_H(a, b) \subseteq M$ be the set of attainable costs. For discrete interfaces and additive independent costs, composition has the formula

$$C_{H \times_B K}(a, c) = \bigcup_b (C_H(a, b) + C_K(b, c)).$$

The sum is the set of pairwise sums. When taking an infimum commutes with addition, one obtains the familiar min-plus composition of value functions. The witness-level formula distinguishes an attained minimum from an unattained infimum. In a higher interface matching paths can have their own cost or can impose a constraint. In that case costs attach to the full pullback witness, including the path. In a shared-capacity model a pair of individually cheap witnesses may exceed the joint bound, and the appropriate constraint must be imposed before taking the infimum.

6.8 Dual views of evidence

An observer can ask for a direct witness of an output, or for a proof that every compatible record has that output. For a map $p : H \to B$ these questions are related to the adjoints to pullback in slices. The dependent sum Σ_p composes maps with p, and the dependent product Π_p expresses choice of a witness over all fibers. An ∞-topos is locally cartesian closed, so both adjoints exist. On subobjects, the existential condition is the image of Σ_p. The universal condition is the corresponding dependent product of propositions.

In particular, a universal predicate can hold at a boundary with no possible record. This is the usual empty-domain convention for universal quantification. The architecture keeps an independent feasibility condition. In an honest network proof, this prevents an impossible input from being silently counted as a successfully supported workload.

7 Causality, schedules, and fair runs

7.1 Event order

Given one finite run, order events on each process by the observed local order. For each delivered message add an edge from its send to its receive. The transitive closure is Lamport's happened-before relation [9]. It is a strict partial order when the directed dependencies are acyclic, as they are for

a run generated by the atomic interleaving model. Two observed events can be incomparable even if a physical clock assigned one a smaller timestamp. Conversely, a timestamp agreeing with the partial order can be chosen by a topological sort.

A consistent cut is a downward-closed subset $I \subseteq P$ of the event poset. If it contains a receive it contains the matching send. If it contains the send but not the receive, the message is pending at the cut, subject to the chosen channel model. The set $J(P)$ of all cuts, ordered by inclusion, has meets and joins given by intersection and union. Both operations preserve downward closure. $J(P)$ is a distributive lattice because the set operations satisfy the distributive laws.

Proposition 7.1 (Reconstructing the causal order)**.** *Let P be a finite event poset. For events $a, b \in P$, $a \leq b$ if and only if every consistent cut containing b contains a. The nonzero join-irreducible cuts are exactly the principal cuts $\downarrow a$.*

Proof. Downward closure gives one implication. For the other take the cut $\downarrow b$. A principal cut cannot be a union of proper subcuts because one subcut of such a union would have to contain its maximal event. Every nonempty cut is the union of its principal cuts at maximal events. If it has two or more maximal events it is a union of proper subcuts, proving the second statement. □

One can therefore recover the poset from the cut lattice up to isomorphism. One cannot recover every channel operation from that lattice: an extra edge whose endpoints were already ordered leaves the poset unchanged. The identity of a message and its payload are labels on the poset, not order-theoretic invariants.

7.2 Vector coordinates for consistent cuts

Partition the events of P into m process chains. A cut I gives the vector

$$\nu(I) = \big(|I \cap P_i|\big)_{i=1}^{m}.$$

Each local component of a cut is a prefix of its chain. Thus ν is an order embedding from $J(P)$ to the product of the integer intervals for local prefix lengths. The image is closed under coordinatewise minima and maxima because intersections and unions of cuts are cuts. An arbitrary tuple of local lengths may not belong to the image: a receive present in one chain might require a larger coordinate on another.

The timestamp of event a is $\nu(\downarrow a)$. In this finite labeled model it can be computed by propagating vector maxima along the local chain and along send-to-receive edges, then recording the current chain position at a. Every principal ideal is the union of the principal ideals of the immediate predecessors and the new event. This proves the formula and the following order test by induction over a topological order.

Corollary 7.2 (Vector order test)**.** *For $a, b \in P$,*

$$a \leq b \quad \Longleftrightarrow \quad \nu(\downarrow a) \leq \nu(\downarrow b)$$

coordinatewise.

A scalar logical clock can preserve strict causal order without reflecting it. Two independent one-event chains give a simple counterexample to reflection by any increasing total ordering that separates the events. The exact vector statement assumes all relevant dependencies have been included. If a shared memory read is causally dependent on a write, a clock scheme that propagates information only along selected channels can miss that dependency. For background on vector time see [11].

7.3 A cubical model of partial execution

A finite poset presents a model where any set of simultaneously enabled events can execute independently. This is appropriate if all constraints are already in the dependency order and if all incomparable actions compatible with a cut really can be run together. Define a subspace

$$\mathcal{Q}(P) = \{x \in [0,1]^P : \Phi(x)\},$$

where the defining condition is

$$\Phi(x): \quad x_b > 0 \;\Rightarrow\; x_a = 1 \quad (a < b).$$

A coordinate is zero before the event, one after it, and strictly between zero and one during it. The vertices are indicator vectors of cuts. At a cut I one can vary the coordinates in a set A of minimal elements of $P \setminus I$. This gives a cube with vertices $I \cup B$ for $B \subseteq A$. Every point of $\mathcal{Q}(P)$ is in one of these cubes, using its completed and in-progress coordinates. The faces agree as coordinate subcubes of the ambient cube.

Theorem 7.3 (Finite causal cubes)**.** *The cubical space $\mathcal{Q}(P)$ is contractible. The saturated directed vertex paths from $\varnothing$ to P are exactly the linear extensions of P. Any two such paths are related by swaps of adjacent incomparable events, each witnessed by an available square.*

Proof. For a maximal event m, replace its coordinate x_m by $(1-t)x_m$, for $0 \le t \le 1$, leaving all other coordinates fixed. No constraint uses m as a strict predecessor. If the new coordinate is positive, its predecessors were already completed. Hence this is a deformation retraction to the subspace $x_m = 0$, canonically $\mathcal{Q}(P - \{m\})$. Iterate to the unique vertex for the empty poset.

A directed edge adds a single minimal remaining event, so a saturated path lists each event after all its predecessors. Conversely, every linear extension gives such a path. To compare two extensions, take the first event in the target extension and move it left in the other one. Every crossed event is incomparable: it cannot be a predecessor of the first target event, and a successor could not have preceded it in the other extension. Adjacent incomparable events can be run in either order from their common cut, furnishing a square. After aligning the first events, repeat on the remaining poset. □

The theorem asserts a fact about the underlying undirected cubical complex and a separate fact about vertex paths. It makes no identification of all directed mapping spaces for arbitrary cubical execution categories. Directed topology retains which coordinates can increase along a path [13]. In particular, the deformation used above may decrease a coordinate. It is an undirected contraction of the space.

Example 7.4 (A missing cube)**.** Three pairwise independent-looking actions can each share one token of a resource with capacity two. Each pair can execute concurrently, but the triple cannot. All edges and square faces of the 3-cube can be present without the interior 3-cell. The underlying space is S^2. Pairwise commutation alone does not authorize a 3-dimensional witness.

7.4 Capacity skeletons and their holes

Let n actions have no causal precedence and let a capacity rule allow at most k of them to be in progress simultaneously. Suppose a completed action releases the capacity and every order of completed actions is allowed. The corresponding space is the k-skeleton of the n-cube. Its i-cells, for $i \le k$, choose i active coordinates

and assign endpoints to the others. The number of these cells is

$$c_i = \binom{n}{i} 2^{n-i}. \tag{7}$$

Proposition 7.5 (Capacity homology)**.** *For $1 \le k < n$, the capacity skeleton is connected, has reduced homology only in degree k, and has free integral homology there of rank*

$$(-1)^k \left(\sum_{i=0}^{k} (-1)^i c_i - 1 \right).$$

It is simply connected for $k \ge 2$. At $k = n$ it is contractible.

Proof. The 1-skeleton is connected by changing one coordinate at a time. For $i < k$ the cellular differentials into and out of degree i are the same as in the full contractible cube. In degrees above k the chain groups vanish. The top homology is a subgroup of the free abelian group of k-cells and is free. Euler characteristic gives its rank. Attaching cells of dimension at least three does not change the fundamental group, so the 2-skeleton has the same fundamental group as the cube. The final assertion is the case of the full cube. □

When $k = 1$ the rank formula becomes $n2^{n-1} - 2^n + 1$. For $n = 2$ it gives one loop corresponding to the two sequential orders, which have no square comparison. The calculation makes clear that the source of the obstruction is a resource constraint, not a deviation from the protocol. If capacity restrictions also forbid some completed action sets, the skeleton formula no longer applies. The remaining cube complex must be calculated from the actual guard.

7.5 Finite schedules with time bounds

A different enrichment attaches a real variable t_i to an event. Use a difference constraint $t_j - t_i \le u_{ij}$ for every directed weighted edge $i \to j$. A lower bound $t_j - t_i \ge \ell$ is the reverse edge $j \to i$ of weight $-\ell$. An interval $[\ell, u]$ adds both. Fixing a time origin can be done by subtracting the origin's time from every coordinate. Pairwise bounds alone leave a simultaneous translation freedom. Temporal constraint networks are treated in [26].

Theorem 7.6 (Time feasibility and tight boundaries)**.** *A finite difference-constraint system has a solution if and only if its weighted graph has no negative cycle. Assume the graph is strongly connected and feasible, and let $d(i, j)$ be its shortest-path distance. The projection to the difference $t_j - t_i$ is the closed interval*

$$[-d(j, i),\ d(i, j)].$$

If all weights are integers, both endpoints have integer-valued realizing schedules.

Proof. Summing the constraints on a directed cycle gives 0 at its left side, so a negative total is impossible. Without a negative cycle, adjoin a fresh source with zero-weight edges to all vertices. The finite shortest-path distances from that source satisfy every edge constraint by the triangle inequality and therefore give a solution.

For the interval, summing on paths gives the two bounds. A schedule $t_v = d(i, v)$ realizes the upper bound, and $t_v = d(j, v)$ realizes the lower bound. Both schedules satisfy the constraints. The solution set is an intersection of affine half spaces and is convex, so interpolation realizes all intermediate differences. With integer weights the realizing distances are integers. □

A compatible family of individual intervals can still give a negative cycle. The failure certificate consists of the cycle and its weight. When no negative cycle exists, the completion set at fixed feasible boundary values is a convex polyhedron. In its ordinary topology it is contractible by straight-line interpolation to any chosen point. This says nothing about the topology of a schedule space that also imposes disjunctive resource choices, such as either i preceding j or the reverse. Disjunction can separate components.

7.6 Inverse limits of finite prefixes

Let T_n be the honest histories through stage n, with truncation $T_{n+1} \to T_n$. Fix the external inputs through every stage. After taking the relevant boundary fibers, an infinite history is a point of

$$T_\infty = \lim_n T_n.$$

For discrete prefix sets with ordinary truncations this means a compatible sequence (t_n). For spaces one requires the whole homotopy coherent family. Taking connected components of the limit can conceal a further issue, as the example below shows.

Proposition 7.7 (Finite-branching existence)**.** *Suppose the T_n are finite nonempty sets and the truncations are total. Then T_∞ has a point. The same assertion holds for the subtree above a fixed prefix if that subtree has nodes at every subsequent level and is finitely branching.*

Proof. Add one root if the first level has multiple elements. A node has arbitrarily long extensions if its descendants occur at arbitrarily high levels. The root has this property. A node with that property has a child with the same property, because a finite union of sets of bounded depths has bounded depth. Recursively choose such a child. The resulting infinite branch is the compatible sequence. Apply the argument to the subtree in the relative assertion. □

Without finite branching, a root can have one child for every positive integer m, with the m-th child supporting exactly m further steps. Then there are arbitrarily long finite histories but no infinite branch. A space of possible natural-number waiting budgets can have precisely this behavior if the budget must be chosen at the first step.

7.7 A coherent lift with an infinite obstruction

Consider variables $x_0, x_1, \ldots \in \mathbb{Z}$ with

$$x_n - 2x_{n+1} = c_n.$$

Every finite initial segment has a solution, obtained by setting its last variable to zero and substituting backwards. If there is an infinite solution, then for all N,

$$x_0 \equiv \sum_{n<N} 2^n c_n \pmod{2^N}.$$

Set $c_{2j} = 1$ and $c_{2j+1} = 0$. The right-hand side converges in $\mathbb{Z}_2$ to $\sum_{j\geq 0} 4^j = -1/3$, which does not come from an integer. Thus no infinite solution exists. This is the elementary difference equation appearing in the description of $\lim^1$ for the tower $\mathbb{Z} \xleftarrow{2} \mathbb{Z} \xleftarrow{2} \cdots$.

The same data can be read as coherence between a chosen countable family of comparison paths. The finite initial segments have lifts, but none of the lifts can be made compatible at all horizons at once. The example does not contradict Proposition 7.7: at every finite stage the unpinned terminal coordinate has infinitely many possibilities. A finite quotient at each horizon can retain all residue constraints and still fail to distinguish an ordinary integer from a 2-adic one without the intended completion condition.

7.8 Obligations and density

In a rooted prefix tree, an obligation can be described by an upward-closed collection $\mathcal{O}$ of prefixes that witness its discharge. For example, after a message has been enqueued, receipt of that message can serve as a finite witness of discharge. An obligation of infinitely many deliveries can be expressed as countably many obligations, one for delivery of at least k messages for each k. Such statements concern all the prefixes of a run.

Theorem 7.8 (A countable fair extension)**.** *Let T be a rooted prefix tree with no terminal nodes. Let $(\mathcal{O}_i)_{i\in\mathbb{N}}$ be a countable family of upward-closed prefix conditions. Suppose every prefix has an extension in each $\mathcal{O}_i$. Then every prefix has an infinite extension meeting all of them.*

Proof. Starting from the given prefix, extend to a node in $\mathcal{O}_0$. Extend by at least one more edge, which exists by the absence of terminal nodes, and then extend to $\mathcal{O}_1$. Repeat. The resulting nested sequence has unbounded length and determines an infinite branch. Each condition already met remains met by upward closure. □

This is an existence theorem. It does not make every admitted run fair. If the next scheduler decision always postpones one particular message, an infinite legal branch can miss an obligation that had an extension at every prefix. A universal liveness proof must restrict schedulers or prove that the protocol itself prevents such a branch.

Example 7.9 (Fairness is a genuine restriction)**.** In a one-process model an enabled request can either be completed or be postponed one time step. The prefix tree admits a run postponing forever. Completion is an open dense condition in the space of runs, but its complement is nonempty. All finite prefixes of the noncompleting run extend to a completing run, so no finite prefix witnesses failure of the eventual-completion property.

7.9 Progress certificates

A safety invariant is usually checked initially and preserved by steps. A progress certificate can use a well-founded set of ranks instead. Let an obligation remain active until a designated completion. Assume that while it is active the rank does not increase. Assume that at any stage of a fair run where the obligation is still active, the obligation eventually completes or the rank eventually decreases strictly. Then every fair run completes the obligation. Otherwise recursively pick the stages of strict decrease, giving an infinite descending sequence.

The eventual-decrease assumption can be justified by weak fairness when a rank-decreasing action remains continuously enabled unless progress already occurs. If it is enabled only repeatedly for brief intervals, weak fairness need not suffice; an appropriate strong fairness assumption or a different progress argument is needed. A local transition rule that leaves the enabled action available describes honesty of individual steps. The quantifier over infinite schedules is what gives eventual decrease.

7.10 When projections lose fairness

An observational quotient can erase actions on which a fairness condition depends. For example, an unbounded sequence of honest internal retry steps can have a constant external trace. If the specification regards that constant trace as indefinitely waiting, a liveness proof that hides the retries needs an explicit relation between the two fairness models. One way to express

that relation is a map of inverse-limit run spaces together with an inclusion of the image of fair runs into the fair part of the target. A pointwise finite equivalence of records gives the limit map only when it is natural in truncations.

Finite deterministic hardware, operating system interrupts, and input queues do not by themselves make the mathematical prefix tree finite at each horizon. A model can have unbounded input strings, real-valued delays, or arbitrarily many available clients at a single stage. If compactness is part of a proof, it must be proved for the actual chosen history space.

8 Observation, knowledge, and internal logic

8.1 The universe of considered runs

Let R be a set of possible records at a specified horizon, possibly subject to an external boundary and an environment assumption. For each observer i fix its view $v_i : R \to V_i$. Its indistinguishability relation is the kernel pair

$$r \sim_i s \quad \Longleftrightarrow \quad v_i(r) = v_i(s).$$

The underlying collection R is essential. A run in which a message was lost is relevant to what an agent knows only if the model admits that run. Restricting the environment assumption can strengthen knowledge without adding a new message to any observed trace. The interpretation follows the runs-and-systems treatment of distributed knowledge [23].

For a proposition $P \subseteq R$, define

$$K_i P = \{r : \{s : s \sim_i r\} \subseteq P\}.$$

Knowledge is an interior operation on the Boolean algebra of predicates on R: it is monotone, preserves arbitrary intersections, satisfies $K_i P \subseteq P$, and is idempotent. These statements follow because each equivalence class is nonempty, and all its elements have the same class. Its dual possibility operation takes the union of classes meeting P. The unit and counit of these inclusions can also be expressed using image and dependent product along the view map.

Proposition 8.1 (Decision by local observation)**.** *A deterministic decision map $d : R \to A$ can be computed from the view v_i exactly when it is constant on $\sim_i$-classes. A proposition P can be decided from that view exactly when $K_i P = P$ and $K_i(R \setminus P) = R \setminus P$.*

Proof. A decision obtained by composing the view with a function on V_i is constant on each fiber. Conversely, constancy gives a unique function on the image of v_i; if a total function on all V_i is needed, specify behavior on views never attained. For a Boolean decision the two preimages of its values must both be unions of classes. This is exactly the two knowledge conditions. □

If the decision is permitted only at some records, first restrict to the domain on which the decision is allowed and determine whether that domain can itself be recognized from the view. A decision map on just the successful cases can otherwise claim more knowledge than the observer has.

8.2 Common knowledge as a fixed point

For observers in a nonempty set G, write $\sim_G$ for the equivalence relation generated by the $\sim_i$, $i \in G$. Its classes are the connected components of the undirected graph whose edges come from any member's indistinguishability. Let $C_G P$ be the union of the $\sim_G$-classes contained in P.

Theorem 8.2 (Common knowledge calculation)**.** *The predicate $C_G P$ is the greatest fixed point of*

$$X \longmapsto P \cap \bigcap_{i \in G} K_i X.$$

Define $X_0 = P$ and $X_{k+1} = P \cap \bigcap_{i \in G} K_i X_k$. Then X_k consists of records from which every record reachable by a chain of at most k indistinguishability edges belongs to P, and $C_G P = \bigcap_{k \geq 0} X_k$.

Proof. A union of $\sim_G$-classes is fixed by every K_i. In particular $C_G P$ is contained in P and is a fixed point. If X is a fixed point, $X \subseteq P$, and from $r \in X$ every one-step indistinguishable record is in X. Induction gives the entire component, so $X \subseteq C_G P$. The description of X_k follows by induction, since the step adds one indistinguishability edge. Every component member is reached by a finite chain, which proves the final formula. □

For finite R, a failure of common knowledge can be certified by a chain from the observed record to one outside P. The shortest length specifies the first failed level of the iteration. A verifier can compute the condition by graph traversal rather than building all nested formulas. The atlas uses this observation to display both first-order and iterated knowledge.

8.3 An acknowledgment ladder

Let the admitted records be $r_0, \dots, r_m$. In r_0 a designated initial notice has not arrived; in r_j with $j > 0$ there have been j alternating deliveries, starting at observer A. View A identifies the pair r_1, r_2, then r_3, r_4, and so on. View B identifies r_0, r_1, then r_2, r_3, and so on. Unpaired endpoints have singleton views. This is a finite observation model of alternating acknowledgments where each additional delivery is invisible to its sender. Let P state that the initial notice arrived.

The union graph is a path with m edges. Both observers may know P at r_m once $m \geq 2$. The k-fold predicate X_k at r_m holds for $k < m$ and fails for $k = m$. No record has common knowledge of P. Every receipt can be honestly generated and every finite prefix with a receipt can be consistent. The obstruction is the collection of other runs with the same individual views.

The example specifies its observations rather than assuming that the existence of acknowledgments determines the relation. A simultaneous public event observed by all agents can split the union graph. A known timing guarantee can eliminate r_0 from the possible records at a later time. Either modification changes the mathematics by changing the views or the underlying universe of runs.

8.4 Restriction of possible records

If $S \subseteq R$, keep the same view maps but restrict their domains. For each $r \in S$, the class of r inside S is the intersection of its former class with S. Consequently

$$S \cap K_i^R(P) \subseteq K_i^S(S \cap P).$$

The inclusion may be strict when all counterexamples in the former class have been removed. There is an analogous inclusion for common knowledge, since deleting records can only shrink generated equivalence classes. An equality holds for every P if S is a union of the corresponding equivalence classes. For an individual observer, the converse follows by taking $P = S$ if a class meeting S is not contained in S.

This sensitivity to R matters in failure analysis. If a verifier compares a crash-stop environment to a perfectly reliable one, the second can make more propositions knowable. A proof that uses the

latter indistinguishability relation does not transfer to the former merely because the implementation code is unchanged.

8.5 Views with higher comparison witnesses

If the record and view objects are spaces rather than sets, a witness of indistinguishability over one view is a path in the view object. For a map $v : R \to V$, use the groupoid object given by its Čech nerve, whose first term of comparisons is $R \times_V R$. Given a proposition $P \hookrightarrow R$, the corresponding knowledge predicate at r asks whether every point of the fiber at $v(r)$ satisfies P. In an ∞-topos this is $v^*\Pi_v P$ interpreted in the slice over R. The dependent product is a proposition because all the fiberwise values are propositions and their dependent product is again a proposition.

The formula has a useful section-level interpretation. At a stage, a claim that the observer knows P quantifies over compatible generalized records, including at refinements of the stage. There is no need to suppose that ordinary points of the topos detect every proposition. For a space over a space, testing the ordinary homotopy fibers suffices.

Proposition 8.3 (Pullback of view semantics)**.** *In a pullback square*

$$\begin{array}{ccc} R' & \longrightarrow & R \\ {\scriptstyle p'}\downarrow & & \downarrow{\scriptstyle p} \\ V' & \longrightarrow & V \end{array}$$

in an ∞-topos, knowledge of a pulled-back predicate along p' is the pullback of knowledge along p.

Proof. Write $f : V' \to V$ and $g : R' \to R$. The base-change map $f^*\Pi_p P \to \Pi_{p'} g^* P$ is an equivalence. Indeed, mapping a test object over V' into the right side is the same as mapping its pullback along p' into g^*P. The pullback square identifies that domain over R with the pullback along p of the test object composed with f. The adjunctions identify the resulting mapping space with that into the left side. Now apply p'^* and use $p'^* f^* = g^* p^*$. □

Arbitrary restriction of R generally does not give a pullback square of this form. It can remove some members of a view fiber while retaining others. The earlier inclusion for sets is precisely the direction expected for such an incomplete restriction.

8.6 Stable knowledge in time

A system with perfect recall supplies maps from each later view to its earlier view. If a proposition P_n at stage n persists, and the considered later runs truncate to considered earlier runs, then individual knowledge of P_n persists as knowledge of its consequence. More precisely, let $u : R_{n+1} \to R_n$ be truncation and let the later view determine the earlier view. If $u^{-1}P_n \subseteq P_{n+1}$, then

$$u^{-1}(K_i^{(n)} P_n) \subseteq K_i^{(n+1)} P_{n+1}.$$

Any later indistinguishable pair truncates to an earlier indistinguishable pair, which proves the inclusion. The same argument applies to common knowledge by truncating a whole indistinguishability chain.

Crash recovery can violate perfect recall. A process may remember that its current replica is initialized without remembering which message justified a prior decision. Restoring the knowledge argument then requires exposing a durable certificate or weakening the proposition to one that still follows from the restored view.

8.7 Geometric assertions and witnesses

In the internal logic of the observation topos, a predicate is a subobject, conjunction is pullback, and existential quantification is image along a projection. Disjunction of predicates is the image of their coproduct in the common ambient object. A finite conjunction, arbitrary disjunction, or existential construction of this kind is preserved by inverse image functors of geometric morphisms. This follows from the preservation of finite limits, colimits, and images established in Section 3. It gives a precise class of local proof obligations that transport to a changed observation site.

Universal quantification has the fiber semantics above but need not commute with arbitrary geometric inverse image. For example, on the real line take the sequence of open neighborhoods $U_n = (-1/n, 1/n)$ of the origin. Each associated proposition has true germ at the origin. Their meet in the lattice of opens is the interior of the intersection, which is empty. Taking the germ of this countable conjunction is false. A universal assertion over a countable constant index can exhibit the same distinction. Thus proof transport for an assertion about all time horizons needs the additional preservation property.

8.8 Logical consequences for diagnosis

Consider an observed boundary b and a conjecture that at least one local rule was violated. A test that proves $\mathrm{fib}_b(H \to B)$ empty refutes the combination of honesty, the observation scheme, the ambient rules, and the recorded boundary. To conclude that a transition was dishonest one must know that all the other premises hold and that the observations were taken from a joint record. Even then the proof may show only a disjunction over possible violations. Knowledge of a particular violation requires the same conclusion on every record compatible with the observer's view. This can be tested by the fiberwise knowledge predicate and is stronger than detecting inconsistency of one proposed reconstruction.

9 Agreement, quorums, and limits of honesty

9.1 What a consensus statement quantifies over

A single-decision task has proposed values, decided values, and a specification relating them. Agreement says no two decisions in one admitted run differ. A validity condition specifies permitted decisions, for example that each decided value is some process's input. Termination says that each eligible process eventually decides under the stated environment and scheduling conditions. In a crash-stop model the eligible processes for termination are normally those that do not crash.

Agreement and validity can be checked as safety properties of finite records. A pair of different decisions is a finite certificate of failure. An infinite run can violate termination while each finite prefix is consistent with future decision. The prefix criterion for progress in Section 7 therefore cannot be replaced by a local agreement check.

9.2 Quorum families and exact resilience

Let N be a finite set of acceptors. A quorum family $\mathcal{Q}$ is a nonempty collection of nonempty subsets of N. We can specify its inclusion-minimal members and take their upward closure. Let $F \subseteq N$ be a set of

failed or unavailable acceptors in the selected window. The availability predicate is

$$\mathsf{avail}_{\mathcal{Q}}(F) \quad \Longleftrightarrow \quad \exists Q \in \mathcal{Q} :\ Q \cap F = \varnothing.$$

This describes which certificates can be assembled in principle from the surviving acceptors. It assumes their replies become usable in the window when applied as a runtime availability claim.

Theorem 9.1 (Dual cut test)**.** *An unavailable set F blocks $\mathcal{Q}$ exactly when it hits every member of $\mathcal{Q}$. Let $\tau(\mathcal{Q})$ be the least size of such a hitting set. Every failure set of size at most f leaves $\mathcal{Q}$ available if and only if $f < \tau(\mathcal{Q})$. For families $\mathcal{Q}_1, \mathcal{Q}_2$, all cross-pairs intersect if and only if every member of $\mathcal{Q}_1$ hits $\mathcal{Q}_2$.*

Proof. Negating existence of a disjoint quorum says that every quorum meets F. The least cardinality of such a set is the first number of failures for which availability can fail. The cross-pair statement is the hitting condition with F equal to a member of $\mathcal{Q}_1$. □

For threshold quorums of size at least q, the least hitting size is $|N|-q+1$. A threshold read family of size q_1 and a threshold write family of size q_2 cross-intersect exactly when $q_1 + q_2 > |N|$. The inequality is sufficient by pigeonhole and necessary because disjoint subsets can be chosen when their sizes sum to at most $|N|$. With nonnegative weighted quorums a strict sum of thresholds above the total weight is sufficient. It need not be necessary if the weights cannot be divided at the threshold values. Weighted replication is classically treated in [20].

Example 9.2 (A single pivotal location)**.** There are five acceptors and all minimal quorums contain a. Cross-intersection can hold even for quorums of size two, but failure of a blocks the family. This is different from a three-of-five quorum, for which any two failures leave a quorum. Intersecting families can have very different failure profiles even when their minimal certificates have similar sizes.

9.3 Quorum evidence as a completion fiber

At a fixed boundary containing the available replies, a quorum certificate can be represented by the set of valid responding subsets. If the witness is a discrete set, distinct subsets are distinct certificates. If comparisons allow adding consistent replies, the witness category can have a terminal object consisting of all available consistent replies. Its nerve is contractible when that full set is permitted as a quorum. These are two different semantics for evidence. The second forgets the choice of a minimal proof when enlarging proofs is allowed. The first retains enough information to compute costs or dependence on particular responders.

The support of the witness map is the availability predicate. Under a topology in which different observers see different replies, local support need not imply a joint certificate for a fixed observation. In particular, reports referring to different epochs cannot be pooled unless the protocol specifies a compatible joint epoch.

9.4 Agreement from a durable ballot invariant

Here is a one-slot crash-stop ballot model. Ballot numbers are natural numbers and each ballot has at most one proposal. An acceptor maintains a promise and its highest accepted ballot with value, if any. It replies positively to a prepare for b only if its promise is at most b. It then raises the

promise to b and reports its prior highest accepted pair. The proposer must collect a phase-one quorum P of these replies before it proposes. It uses the value from the largest reported accepted ballot when one exists. Otherwise it uses an input. An acceptor can accept a proposal for b only if its promise is at most b, and then raises its promise to b. A value is chosen when accepted at one ballot by a phase-two quorum. For recovery the two acceptor fields must persist, or an equivalent epoch discipline must enforce their effect.

Theorem 9.3 (Cross-quorum safety)**.** *Assume every phase-one quorum intersects every phase-two quorum and the rules above hold. All chosen values agree, and each is an input.*

Proof. Fix a chosen pair (b, v) and its accepting quorum Q. Suppose a larger ballot c proposes a different value, and choose the least such c. Its phase-one quorum contains an acceptor $a \in Q$. The acceptance at b by a occurred before its positive prepare reply for c, because the promise from that reply would forbid a later acceptance at b. Thus the prepare record at a reports a ballot at least b. Every accepted pair reported before the unique proposal at c has ballot strictly below c: higher ones would prevent a positive prepare reply and an acceptance at c would require that proposal first. The maximal reported ballot is therefore in $[b, c)$. At b its value is v by uniqueness of the proposal; at a ballot strictly between b and c its value is v by minimality. The proposal at c must use v, a contradiction. Given two chosen ballots, apply this argument to the smaller one. For validity, induction on the ballot number traces every reused value to an earlier proposal, with a fresh value required to be an input. □

This is the safety argument behind the prepare/accept structure of Paxos [18]. It uses cross-intersection, not a claim about the order of messages sent by distinct proposers. It also uses persistence of acceptor state, which is separate from the intersection test. If the same acceptor can respond after losing its promise, a combinatorial intersection can be present without transmitting the information the proof needs.

9.5 A sufficient progress condition for the ballot model

Fix a live proposer with a ballot b. Suppose that after some stage it can exchange messages with all members of a phase-one quorum and a phase-two quorum. Suppose their outstanding promises are at most b and no larger ballot interferes from that stage onward. If the proposer sends its prepare, collects those replies, sends its proposal, and collects the accepts under the delivery and scheduling assumptions, the value becomes chosen. The proof is a finite chain of enabled obligations, each discharged by the environment guarantee.

An eventual leader condition is one way to arrange this premise. With time bounds that eventually hold, there can be enough time to complete a stable ballot. A failure detector can supply a different type of assumption. See the partial synchrony models of [16] and the failure detector framework of [17]. The hypotheses in the previous paragraph are stated as sufficient conditions, so no particular leader election algorithm is implicit in the theorem.

9.6 Configuration change

With two successive configurations, let $\mathcal{P}_{\text{new}}$ be the new phase-one family and $\mathcal{Q}_{\text{old}}$ the old phase-two family. If some member of one is disjoint from a member of the other, Theorem 9.3 does not justify reusing

the same slot across that change. An old chosen value can be invisible to the new prepare. One repair is to require a transition certificate that includes evidence from both configurations; another is to move a durable decision record along an explicit interface before using the new configuration for the slot.

Proposition 9.4 (Two-configuration bridge)**.** *Suppose a transition stores a certificate identifying the chosen value of every previously decided slot in all nodes of a newly required intersection set C. Assume every new prepare for an old slot observes a member of C, and the new rule uses the certified value whenever present. Then no new prepare for those slots proposes a different value.*

Proof. Every such prepare receives at least one report from a member of C. The report contains the previously chosen value and the rule requires using it. The conclusion does not depend on which new acceptors form the phase-two quorum. □

The premise requires a bridge strong enough to capture all decisions preceding the transition. In a live implementation one must also handle decisions concurrent with reconfiguration. Freezing old writes or using a joint phase are ways to make the boundary precise. For an example in a consensus protocol, see the membership change discussion in [19].

9.7 The deterministic asynchronous obstruction

In the asynchronous message-passing model used by Fischer, Lynch, and Paterson, messages can be delayed and process steps can be scheduled at arbitrary relative speeds. There is no bound on delivery time and a process can stop permanently. Even with reliable communication as defined in that model and at most one process crash, a deterministic consensus protocol satisfying the specified agreement and validity requirements has a possible nondeciding admissible execution [15]. That theorem concerns the universal termination quantifier. It permits decision on many schedules and does not forbid the safety argument above.

A finite compatibility proof for each reported decision cannot remove the nondeciding schedule. In the language of this manuscript, the honest run space and its decision subspace can be distinct even when there are no inconsistent pairs of decisions. To prove universal termination one must restrict the run space or use a specification with a different premise, such as partial synchrony or a probabilistic progress statement with its own quantifiers.

9.8 A partitioned register obstruction

For an independent elementary example, consider a read/write register initially zero. The external interface requires linearizability: completed operations admit a total order respecting the real-time order of nonoverlapping operations, and reads return the last written value in that order [21]. Suppose two locations can be permanently partitioned, each accepting local client operations. Assume every local operation at each live location must terminate during the partition.

Proposition 9.5 (Read after a separated write)**.** *Under these assumptions the register cannot be implemented, even with honest computers.*

Proof. In one run, perform a write of 1 at the first location during the partition. It terminates by assumption. Then invoke a read at the second location. In a second run omit the write and reproduce the

reader's initial state, time of invocation, local steps, and observations. The partition delivers no distinguishing message, so the reader's view can be the same through its response. A guaranteed decision from the same view must be the same. Linearizability requires the first read to return 1 and the second to return 0, a contradiction. If the process uses choices, the argument uses matching choices in the two runs for a guarantee quantified over every run. □

This is a special case of the relationship between availability and strong consistency under partitions studied by Gilbert and Lynch [22]. It is also a direct application of Proposition 8.1. Both local traces can be honest and individually satisfiable. The two runs are separated by a fact that the reader has no way to observe.

9.9 Classifying distinct failure reports

A finite network diagnosis can now separate several mathematically precise reports. An invalid transition refutes a conformance predicate. A disjoint pair of certificates refutes an intersection hypothesis. A hitting set refutes quorum availability in a failure window. A matching cycle refutes compatibility of proposed records. A nondeciding run refutes a universal progress statement. These reports differ in the types of witnesses they require. None follows merely from saying that a process was slow or that a local log did not contain an acknowledgment.

10 From local rules to service contracts

10.1 Consistent snapshots as completion problems

In a reliable point-to-point FIFO system, each process has a local prefix at a consistent cut. The state of a channel at that cut consists of messages sent in the sender's prefix and not yet received in the receiver's prefix, ordered as at send. If initial channel contents are present, include them as initial send occurrences ordered before the subsequent sends. If messages can be omitted, include omission events that remove them. Thus reconstructing a snapshot depends on the selected channel model.

Proposition 10.1 (Reconstruction with complete identifiers)**.** *Suppose all delivered messages have identified send occurrences, the event poset includes all local order and send-to-receive order, and the given local prefixes determine a cut in this poset. For reliable FIFO channels their in-transit contents are determined uniquely by the cut.*

Proof. For each channel, put its sent occurrences in send order and remove those received in the cut. Downward closure ensures every removed occurrence was sent in the cut or was present initially. FIFO implies the received occurrences are an initial segment of the sender's sequence. The remaining ordered suffix is exactly the buffer. □

The proposition reconstructs a record once its cut is specified. Chandy and Lamport's marker algorithm is a distributed way to obtain a consistent record without a shared clock under FIFO and delivery assumptions [10]. When a process records its state it sends a marker on every outgoing channel before further application messages. On a channel where the receiver has already recorded its state, application messages received up to the marker belong to the recorded channel state. FIFO

makes that marker a boundary between messages sent before and after the sender's recording.

One cannot use that same boundary without a replacement for FIFO. For instance, if a later marker can overtake an earlier application message, observing the marker does not determine whether that earlier message was in flight at the cut. An alternative can retain explicit sequence information or use a different snapshot algorithm. In this framework it is a change to the overlap constraint relating the two local prefixes, not a change to the meaning of honesty.

10.2 Replicated semilattices

Let a replicated state take values in a join-semilattice $(L, \vee)$ with a bottom element. Suppose every local update joins a contribution u to the current state, and every received state is joined as well. The model may allow repeated delivery, reordering, or delayed delivery. Joins are associative, commutative, and idempotent.

Proposition 10.2 (Convergence after a finite update set)**.** *Assume there are finitely many contributions, each comes from a permitted update, and every live replica eventually incorporates every contribution, directly or through merged states. Then every live replica eventually equals the join of those contributions and remains there if no new contribution is made.*

Proof. Induct on steps to express each state as a join of some subset of the permitted contributions, counting the common initial state as one if necessary. A merge takes the union of the subsets in this representation. After a live replica incorporates all contributions its value is their total join. Later duplicates leave it unchanged by idempotence. The largest of the finitely many incorporation stages exists for each replica. □

This proof has a safety half and a progress half. Safety is the representation of states as permitted joins. The progress half is dissemination. In a permanent partition the first holds even when the second fails. State-based convergent replicated data types use this structure in more general settings [24]. A complete specification also needs to say what a query means while different replicas hold different joins. Convergence is compatible with many different interim observation policies.

Example 10.3 (Grow-only set)**.** The states are subsets of a universe of request identifiers, the join is union, and an update inserts one identifier. An observed set S is evidence that those identifiers were incorporated. It does not show that an unlisted request has never occurred at another replica. Under complete dissemination of a finite update set, every live replica converges to the same final subset. An observer asking whether a request *will* appear needs the corresponding progress assumption.

Example 10.4 (A removal needs more state)**.** Suppose one replica sees an addition of x and then removes x, while another has only seen the addition. If the first transmits just the resulting empty set, a union merge at the second resurrects x. A state that also records the identity of the removed addition can distinguish that history from one in which no addition occurred. Local application of an honest rule to the smaller representation did not retain enough information for the desired merge. The corrected interface makes the cancellation evidence explicit.

10.3 Resource conservation with local rights

Suppose a service has K units of capacity shared among processes. Requiring each process to keep its own count at most K does not impose the global bound. Two disjoint views can each honestly permit an operation that consumes the last unit. One way to make the bound local is to allocate disjoint rights. Use a fixed set L of K unit identifiers. Each identifier is held by one process, in transit with a unique transfer record, or consumed.

Theorem 10.5 (Conservation in a transfer network)**.** *Suppose initially every identifier of L has exactly one owner. An allowed transfer removes an identifier from the sender's holdings, creates exactly one in-transit token for it, then replaces that token with a holding at the recipient. An allowed consume replaces a held identifier with a consumed marker. No other transition changes these coordinates. Then no identifier is consumed twice and the total consumption is at most K in every finite prefix.*

Proof. For each identifier, the number of its occurrences among holdings, transit, and consumed markers is one initially. Each listed transition removes one occurrence and adds one occurrence. Frame conditions preserve all other identifiers. Thus the invariant holds for all prefixes. A consumed identifier cannot be held again by any transition in the specification. The bound follows by counting consumed identifiers. □

Durable evidence is needed when holding records and channel updates can be separated by a crash. For example, if a sender deletes a right durably but queues its transfer only in volatile memory, recovery can strand the right. The at-most-one invariant may still hold while the no-loss invariant fails. If a retry can create a second token, the safety invariant itself fails. The honest specification should describe transfer identity and durable state at the same atomic granularity as the conservation argument.

Liveness is a different property here as well. A requester might lack local rights while unused rights remain at another machine. Eventual acquisition requires a transfer policy and delivery hypotheses. No local conservation argument determines which request wins. At the interface one can expose ownership, current availability, and admissible transfer witnesses as separate coordinates.

10.4 Atomic commitment and a prepared state

In a one-shot transaction, each participant can vote prepared or abort. A commit requires all participants' preparation. A two-phase coordinator collects votes, records a decision durably, and communicates the decision. A participant that has already voted prepared cannot unilaterally abort if a commit may have been recorded and the participant has no way to learn the coordinator's state. Consider two runs with the same prepared view at that participant. In one, the coordinator recorded commit before crashing; in the other, a different participant voted abort. An unconditional local choice of commit or abort cannot be correct in both runs.

This is a knowledge argument about the prepared view, not a proof that every two-phase implementation has exactly the same behavior. A replicated decision authority can maintain a different observation contract. Gray and Lamport compare two-phase commit with a consensus-based commit protocol [25]. A larger decision layer

changes the conditions for learning and recording a transaction outcome, while the all-participants preparation condition remains distinct from replication of the decision.

Proposition 10.6 (A commit certificate is a pullback)**.** *Let $P_i \to B$ be the space of preparation certificates from participant i over a common transaction descriptor B. The space of commit-eligible tuples over a fixed b is $\prod_i \mathrm{fib}_b(P_i \to B)$. To authorize an observable commit, a separate decision record can be required over this product.*

Proof. The combined preparation object in the slice over B is the slice product of the P_i. Pullback along b preserves this product, giving the displayed formula. A decision record over it is then another map of objects; its existence does not follow from the product unless the decision rule supplies it. □

An abort can be modeled by a different family of witnesses, such as a rejection or an admissible coordinator decision. Timeout alone may give knowledge that the local wait expired but not knowledge that preparation was impossible. This distinction is visible when specifying the descriptors for prepared, decided, and learned outcomes.

10.5 Reading a replicated log

Let a service log be an ordered sequence of operations. Once a prefix is chosen, deterministic application of that prefix gives a state. For a read to be linearizable relative to the specified write acknowledgments, it is not enough that the reader holds a valid log prefix. The reader must know that its read can be placed in a linearization respecting operations completed before the read began. A prefix can be valid but stale. For example, reading the empty log after a write has completed elsewhere may return a state impossible in any linearization of those two completed operations.

A read certificate can therefore carry a frontier, epoch, or acknowledgment witness in addition to the returned value. Its semantics is a boundary fiber over operation invocations and responses. A rule that removes the witness when presenting the user-facing value can still retain it in the proof object. If the rule removes the frontier before checking a cross-epoch comparison, it can admit inconsistent combinations of otherwise honest results.

10.6 Two implementations of a request ledger

Consider a service with request keys and an append-only set of applied requests. At each key, a stored record contains a payload and one deterministic result. The specification allows a retry of an already applied key to return the same result, but it forbids a second different payload at that key. There are three conceptually different implementations of the interface.

1. One machine applies a durable transaction per new key. Its atomic log entry stores both payload and result. The local record at a retry can be mapped directly to the existing entry.

2. Several machines can use an agreed per-key decision before returning a committed response. The response certificate includes the decision epoch and its chosen value. The safety proof uses cross-quorum agreement.

3. Independent replicas can accept insertions into a grow-only set of $(\mathrm{key}, \mathrm{payload})$ pairs. They converge after dissemination, but a key with two distinct payloads persists as two different pairs unless the interface treats such

a pair as a separately declared conflict.

The third implementation has the same observable behavior as the first only on a restricted input class, for example if an external invariant supplies one payload per key. To compare them in a context that can submit two different payloads at one key, include that input in the boundary. The completion supports then differ. If the input class forbids the conflict, comparing a latency or progress guarantee still requires the delivery assumptions for the two implementations.

10.7 A reusable proof obligation for idempotence

Suppose an implementation stores a durable map $D : K \rightharpoonup P \times R$, with keys K, payloads P, and results R. An apply step at an unused key atomically adds a pair. A repeat at a used key with matching payload returns the same result, and a repeat with a different payload is rejected. Crashes preserve D. Then by induction over steps D is a monotone partial function under graph inclusion. At each key there is at most one successful application, and every successful response at that key agrees with the pair in D.

For a network of such replicas one needs a global premise. If every key has a unique durable authority at all times, the union of the per-authority maps is a partial function. If authority migrates, the old map at that key must be included in the transfer boundary. If replicas concurrently decide keys, intersections or another agreement mechanism must establish that distinct records for the same key cannot both become successful. The individual induction is still valid; it simply has a smaller domain.

10.8 An implementation-to-proof worksheet

A complete contract can specify the following maps in order.

1. A raw record to its observation boundary, with a statement of which event identifiers and failure cuts are visible.
2. The honest subobject of raw records, given by a stable family of conformance checks.
3. A witness object over each accepted output and the map forgetting its private evidence.
4. A tower of prefixes compatible with all the boundary projections.
5. The allowed fair runs and the condition they must eventually witness.

The worksheet can be used for an existing system. For example, if a failure is observed, test its boundary fiber before changing the scheduler model. If the fiber is empty, identify an inconsistent pair, cycle, or missing field. If it is inhabited, ask whether every completion meets the safety condition. If safety is satisfied and a response is late, examine which fairness hypothesis was used to infer eventual completion. The mathematical type of the question determines the appropriate certificate.

11 A calculus of observable conformance

An implementation can expose the data that decide whether a transition is honest, or it can expose a result that is compatible with several private histories. These possibilities have a categorical separation test. The test works for propositions on sheaves of higher records and identifies ex-

actly which propositions survive a chosen boundary map.

11.1 The boundary kernel pair

Let $p : X \to B$ be the observation of an ambient record. Replace B by the image of p so that p is an effective epimorphism. Let $R = X \times_B X$ with projections r_1, r_2. The equality $r_1^* P = r_2^* P$ for a subobject $P \hookrightarrow X$ means that whenever two records have equivalent boundary values, the induced propositions agree. This condition concerns all stages. If observations have automorphisms, a point of R includes the matching path.

Write $\exists_p$ and $\forall_p$ for the adjoints to pullback on subobjects. The first is the image of composition with p. The second is induced by the dependent product. Define

$$P^{\mathrm{poss}} = p^* \exists_p P, \qquad P^{\mathrm{cert}} = p^* \forall_p P.$$

The first marks records whose boundary permits a witness locally. The second marks records whose boundary forces the proposition in every compatible record.

Theorem 11.1 (Exact test for observable conformance)**.** *Let $p : X \twoheadrightarrow B$ be an effective epimorphism in an ∞-topos. For $P \hookrightarrow X$, the following are equivalent.*

1. *P is the pullback of a subobject of B.*
2. *$r_1^* P = r_2^* P$ in $X \times_B X$.*
3. *$P = P^{\mathrm{poss}}$.*
4. *$P = P^{\mathrm{cert}}$.*

The subobject on B is unique. It is both $\exists_p P$ and $\forall_p P$. In general P^{poss} is the least pullback predicate containing P and P^{cert} is the greatest pullback predicate contained in P.

Proof. For every $Q \hookrightarrow B$, $p^* Q \to Q$ is an effective epimorphism by base change, so $\exists_p p^* Q = Q$. Thus pullback is injective on subobjects and reflects their order. The other adjunction then gives $\forall_p p^* Q = Q$: the counit gives an inequality after pullback, which reflects the inequality, and the unit supplies the reverse one. The unit and counit of the two adjunctions give $P^{\mathrm{cert}} \subseteq P \subseteq P^{\mathrm{poss}}$ and their claimed extremal properties. This proves equivalence of the first, third, and fourth conditions.

Any pullback predicate satisfies the second condition by commutativity of the kernel-pair square. Conversely, by base change of images, $p^* \exists_p P = \exists_{r_2} r_1^* P$. Under the second condition this is $\exists_{r_2} r_2^* P = P$, since r_2 is a base change of p and hence an effective epimorphism. The third condition follows. □

This result specializes to the partition test of Proposition 8.1. To test whether honesty is determined by an observation, take X to include raw records that can fail the predicate. The smaller domain of honest records remains appropriate for analyzing properties of honest executions, such as which delivery order was used.

11.2 Certainty and unresolved observations

With $H \hookrightarrow X$ the honest subobject, let $H_+ = \forall_p H$ and $H_- = \forall_p \neg H$, where $\neg$ is the pseudocomplement. Their intersection is bottom when p is an effective epimorphism. The right adjoint preserves meets, $H \cap \neg H$ is bottom, and $\forall_p$ sends bottom to bottom in this case since pullback reflects order. The disjunction of these two certainty predicates need not exhaust the boundary. On sets the remainder consists exactly of boundaries with both an honest and a nonhonest completion. On a site the two certainty predicates can hold only after further restrictions.

Example 11.2 (Crash between result and durable state)**.** A write attempt has a raw

record field marking whether the new value is durable. An exported record retains only the response. Suppose both a correct execution that persists the write before acknowledging and a raw record that acknowledges first have the same external response. The two records give a point of the boundary kernel pair on which the conformance predicate disagrees. The boundary permits persistence but cannot certify it. Adding the durable transaction identifier and a verifier for its pre-response placement can change the kernel-pair test.

Example 11.3 (A status message from a reliable copy)**.** An observer sees the result of a local durable read. If the model's scope is the local copy, the returned value may determine the property of that copy. For the property that all live copies hold the value, a remote copy's state can vary inside one fiber. A claim of agreement needs a stronger observation contract or a protocol invariant implying it on every record of the fiber.

11.3 Composing observation maps

For $X \xrightarrow{p} B \xrightarrow{q} C$, the identity $(qp)^* = p^*q^*$ implies

$$\exists_{qp} = \exists_q \exists_p, \qquad \forall_{qp} = \forall_q \forall_p.$$

These equations hold up to the canonical adjunction identifications. A contract export that first hides local state and then hides an epoch can therefore compute its possible or certain predicates stage by stage.

If both maps are effective epimorphisms, a predicate is observable at C exactly when it is observable at B and its unique descended predicate there is observable at C. To prove this, factor any pullback from C through B. Conversely, apply the two pullback identities in succession. This gives a compositional test for retention of conformance when a network publishes progressively smaller views.

11.4 Possibility under independent and shared witnesses

Let $X_i \to B$ be a finite collection of maps of records to a common boundary. For each i, let $P_i \hookrightarrow X_i$ be a local certificate. The pullback product of these certificates over B has image the intersection of their images. Factor each $P_i \to B$ into an effective epimorphism followed by a monomorphism. Over the intersection of the images the factors are effective epimorphisms. Their finite product is an effective epimorphism because it is a composite of base changes of such maps. Its image is the intersection.

This argument uses independence over the joint boundary B. If two record types must match a private receipt m as well, use $B \times M$ or the relevant bundle $M \to B$ as the interface before taking images. For example, the only receipt supporting the writer may be m_0, and the only receipt supporting the reader may be $m_1 \neq m_0$. Both observations individually permit the same public value b. Their images on B meet at b, but their images over the receipt interface are disjoint.

11.5 Cut observation and a uniform progress bound

Suppose the prefix sets T_n are finite and discrete and write $R = \lim_n T_n$ with its inverse limit topology. The ambient product is compact, and consistency conditions on successive coordinates are closed. Hence R is compact. A condition witnessed by a prefix is a union of cylinder sets, where the cylinder for t_n consists of runs restricting to t_n.

Theorem 11.4 (Uniform bound on a compact family)**.** *Let $F \subseteq R$ be compact and suppose every run in F eventually has a*

prefix witnessing a fixed success condition. Then some finite horizon suffices for all runs in F.

Proof. The cylinders for successful prefixes cover F. A finite subfamily covers F by compactness. Take the maximum of the lengths of the corresponding prefixes. By that horizon every run in F has already witnessed success. □

The compactness hypothesis is substantive. In the postponement example of Section 7, the set of all runs that eventually complete is not closed in R. Runs completing after m waits converge to the run that never completes. Every run in the fair subspace can finish, but there is no uniform stage by which it does so.

11.6 Composition of progress obligations

For a wiring of finitely many networks, suppose a global run projects to an admitted local run of each component. Let each local progress guarantee state that a predicate P_i is witnessed by a prefix, and assume projection sends fair global runs to fair local runs. If the local witnesses persist and refer to one fixed joint boundary, then every fair global run has a prefix witnessing all P_i. For each component take its finite witness time in the projected run. Choose a finite global prefix including all those projected prefixes. Persistence gives the simultaneous witnesses. Lifting from local prefix lengths to a finite global prefix is automatic for a subsequence projection on a linear run when each witnessed local step comes from a finite global step.

An output obligation deduced from the combined witnesses requires a transition or rule that accepts them jointly. Two receipts might refer to different transaction epochs if the boundary does not constrain them. Even in one epoch, local fairness need not follow from a scheduler that is fair on the composite under some other definition. A process may lose eligibility whenever another takes a shared resource. Stating fair projection as a hypothesis makes this failure detectable.

12 Exact methods for the case atlas

The following sections analyze finite specifications of honest operations. A case names the local inputs and observations, gives the mathematical constraints, and records enough intermediate values to reproduce the result. The cases concern different invariants. A causal schedule count should not be compared directly with a count of components of a groupoid, and an availability polynomial is a statement about a model of failures rather than a measured service level.

12.1 Dynamic programming on causal cuts

For a cut $I \in J(P)$, let $p(I)$ be the number of directed paths from $\varnothing$ to I, and let $q(I)$ be the number of paths from I to P. The recurrences are

$$p(\varnothing) = 1, \quad p(I) = \sum_{a \in \max I} p(I - \{a\}),$$

$$q(P) = 1, \quad q(I) = \sum_{a \in \min(P \setminus I)} q(I \cup \{a\}).$$

Removing the last event of a prefix or the first event of a suffix partitions the corresponding paths. Every full schedule passes through one cut of each size, so

$$\sum_{\substack{I \in J(P) \\ |I| = k}} p(I) q(I) = p(P). \tag{8}$$

The number of schedules with event a at position $k+1$ is the sum of $p(I)q(I \cup \{a\})$

over all cuts I of size k at which a is enabled. This is an exact count of total orders, without a scheduler probability assumption.

The earliest position of a is $|\downarrow a|$, and its latest position is $|P| - |\uparrow a| + 1$. For the first, schedule all strict predecessors before a and all remaining events after it. The strict predecessors are a cut, and the immediate extension by a is a cut, so these orders exist. Dually, order all events except a and its strict successors before a to achieve the latest position. Any schedule must obey the indicated bounds. The width is the maximum size of an enabled set at a cut: enabled sets are antichains, and each antichain is enabled after executing the union of its strict predecessor cuts.

12.2 Integer and field coefficient certificates

For a finite 2-complex with one vertex, r loop generators, and s attaching words, let M be the $s \times r$ matrix of exponent sums in the words. The cellular cochain map $d : C^1 \to C^2$ is multiplication by M. This follows from the degree of the attaching map in each circle after collapsing the others [6]. If a set of edge labels is pinned, delete their columns and subtract the known contribution from the required right-hand side. For a relative problem with the vertex also fixed, C^0 is zero. In the absolute problem, $C^0 = A$ and $d^0 = 0$.

Over $\mathbb{F}_p$, row reduction of M with the identity appended produces a matrix L with LM in reduced row echelon form. Applying the same row operations to the residual vector c gives Lc. A zero row of LM with a nonzero entry of Lc is a certificate of impossibility. The corresponding row w of L satisfies $wM = 0$ but $wc \neq 0$. If no such row exists, setting the free coordinates to zero gives one solution, and varying them produces the full affine space of lifts. In dimension two every element of C^2 is a cocycle. Thus for an $s \times r$ matrix of rank d the dimensions of H^1 and H^2 over the field are $r - d$ and $s - d$. With a fixed root frame the associated completion space for $n = 1$ is discrete; without it there is also the automorphism group $H^0 = A$.

Over $\mathbb{Z}/\ell\mathbb{Z}$, division by a nonunit cannot be used in this way. If integer row and column operations give $UMV = \operatorname{diag}(d_i)$ with U, V unimodular, the equation reduces to one congruence $d_i y_i = (Uc)_i$ per nonzero diagonal entry and a zero condition on each remaining row. Each such congruence is solvable if and only if $\gcd(d_i, \ell)$ divides its right-hand side, in which case it has exactly $\gcd(d_i, \ell)$ solutions. Free columns contribute ℓ choices each. This follows by taking the kernel and image of multiplication by d_i in the cyclic group, and by invertibility of the unimodular transformations modulo ℓ. It includes coefficients detecting torsion that a chosen prime field might miss.

12.3 Weighted closure of discrete ports

Let A be a finite label alphabet. A relay with one input and one output has a matrix M whose entry M_{ab} counts internal records with those port values. If a record has a multiplicative weight in a commutative semiring one can use the corresponding weighted sum instead. Wiring two relays forms a pullback along the shared set A. Distributing the finite sums gives

$$(M_1 M_2)_{ac} = \sum_{b \in A} (M_1)_{ab} (M_2)_{bc}.$$

For a cycle, identifying the final and initial ports gives the trace of the matrix product. A specific interface label fixes a diagonal entry of the cyclically ordered

product beginning at that interface. At a chosen edge i, the number of full completions realizing $a \to b$ there is $(M_i)_{ab}(R_i)_{ba}$, where $R_i = M_{i+1} \cdots M_k M_1 \cdots M_{i-1}$. The last factor counts the return paths to the pinned input. Summing over (a, b) gives the trace again, providing a check on the edge marginals.

If every row and column of every relay has some nonzero entry, every single port label can occur locally. The trace can nevertheless be zero. For example, let all the relays be permutations and let their product be a fixed-point-free permutation. Its entries describe a global transport on the label set with no fixed label on the cycle. This is a discrete version of the transport obstruction in Section 5. The atlas contrasts it with positive counts that have the same local nonemptiness tests.

12.4 Failure profiles from quorum antichains

For a minimal quorum family $\mathcal{Q}$ on n nodes let a_k be the number of sets of k survivors that contain at least one member of $\mathcal{Q}$. Under independent survival with the same probability p, the exact availability probability is

$$\sum_{k=0}^{n} a_k p^k (1-p)^{n-k}.$$

Each term is the probability of one survivor set of size k summed over the allowed sets. For correlated failures the right calculation sums the actual joint probabilities of the individual survivor sets. In particular, a whole region failing can select one specific set of crashes. An independent-node polynomial cannot supply that region's probability.

The minimal blocking sets are the inclusion-minimal transversals of the quorum antichain. They can be enumerated by testing every subset of N. The cross-intersection profile records, for each j, the number of pairs of minimal certificates with j nodes in common. All cross-pairs intersect exactly when the $j = 0$ coefficient vanishes. Testing only minimal families is enough because adding nodes to either certificate cannot remove an existing intersection. Conversely, availability is possible whenever one minimal certificate survives, so the same antichains determine both profiles.

12.5 A finite criterion for fair noncompletion

Fix a finite transition system with initial state s_0 and an absorbing class of states in which an obligation is complete. Each transition can carry a subset of labels from a fixed set Λ. Admit as fair those infinite runs in which every label in Λ occurs infinitely often. This is a stated model of fairness. A fairness rule conditional on future enabling must first be translated into appropriate acceptance data before this test applies.

Theorem 12.1 (Finite recurrence certificate)**.** *There is a fair run from s_0 that never completes if and only if, in the directed subgraph of noncomplete states, a reachable strongly connected component contains a directed cycle and internal edges collectively carrying every label in Λ.*

Proof. In an infinite run in a finite graph, the vertices visited infinitely often lie in one strongly connected component. After some stage the run uses only such vertices. A recurrent edge exists and belongs to a directed closed walk in that component. If a label occurs infinitely often, some edge carrying that label recurs, since the graph is finite. Thus the component has the required edges and a cycle.

Conversely, reach the component without visiting a complete state. Choose an edge of each required label. Use strong connectivity to link these edges by internal paths into one closed walk. If the list of labels is empty, use the assumed cycle. Repeating the closed walk gives a fair noncompleting run. □

The condition is a specialization of the automata-theoretic approach to verification [27]. A proof of universal fair termination can use the absence of the recurrent component. Fair inhabitation still requires an admitted fair run. In the examples below the outcome state has a self-loop with all labels. Then any finite path to that state extends to a fair run, making nonvacuity explicit.

12.6 Capacity as a family of permitted faces

For a set $L = \{1, \dots, n\}$ of independent actions, let Δ be a simplicial complex on L including all singletons. Interpret $A \in \Delta$ as permission for the actions in A to be simultaneously active. If an action is outside A it is either completed or unstarted. The resulting execution space is

$$X_\Delta = \bigcup_{A \in \Delta} [0,1]^A \times \{0,1\}^{L \setminus A}.$$

For the complete family this is the full cube. For $\Delta = \{A : |A| \le k\}$ it gives the capacity skeleton of Proposition 7.5. The construction is the real moment-angle complex of Δ; see the more general polyhedral product theory in [28, 29].

For a subset $S \subseteq L$, let $\Delta_S = \{A \in \Delta : A \subseteq S\}$. We use reduced simplicial cochains with their augmentation in degree -1; in particular $\Delta_\varnothing = \{\varnothing\}$ contributes one group in that degree.

Theorem 12.2 (Active-set cohomology)**.** *For any commutative coefficient ring R,*

$$H^k(X_\Delta; R) \cong \bigoplus_{S \subseteq L} \widetilde{H}^{k-1}(\Delta_S; R)$$

as R-modules. This is an additive statement.

Proof. Use cellular cochains of the cubical complex with the product orientation. In the cochains of each interval take degree-zero generators $1 = v_0 + v_1$ and $x = v_1$ and degree-one generator $y = e$, where v_i is dual to the vertex i and e to the directed interval. The coboundary has $d1 = 0$, $dx = y$, and $dy = 0$. For a permitted active set A, cubical cochains with active set A have basis the products with y on A and either 1 or x elsewhere. The basis change is invertible over $\mathbb{Z}$ and hence over R.

Index a product by the set S of all coordinates at which either x or y is used and by its active subset $A \subseteq S$. Its differential replaces one x by y if the enlarged active set remains permitted. This preserves S, and the sign is the parity of the number of earlier y-coordinates. On fixed S these are exactly the augmented simplicial coboundary signs on Δ_S, with cubical degree $|A|$ corresponding to simplicial degree $|A| - 1$. Thus the cellular complex is a direct sum of the shifted augmented simplicial complexes. Taking cohomology gives the result. □

Over $\mathbb{F}_2$ the atlas computes the resulting Betti numbers in two ways. The cubical chain matrix has one column per cell, with each active coordinate contributing its two endpoint faces. The induced-subcomplex calculation instead groups contributions by S and applies Theorem 12.2. Agreement of the two computations checks the degree shifts and the enumeration of allowed faces. A nonzero contribution from some S gives

a certificate that the restrictions to those actions cannot all be collapsed through the permitted cubes.

12.7 The role of computed examples

A finite computation can check a specified integer, rank, or certificate. It cannot establish a theorem about every network with a given protocol name. The arguments above establish the formulas for their general classes. A case in the atlas supplies the actual data for a substitution into those formulas, or for a separate small exhaustive calculation. This is why each case reports its presentation and the intermediate data that distinguish a positive instance from a negative one.

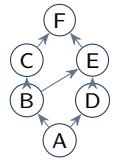
The case labels refer to abstract network roles. For example, a five-replica quorum case can be interpreted as five distinct machines, but the calculations do not assume that such a physical deployment has been observed. Times are exact values in specified units. Counts of witnesses are cardinalities, except when an action groupoid has a stated groupoid cardinality.

13 Atlas of causal schedules

The arrows in these diagrams are the covering dependencies of each event poset. Tables give rank counts, full-schedule counts, position profiles, and the cube census from the enabled sets. All counts refer to distinct named events. See Section 12.1 for the recurrences.

13.1 C01. 2 by 3 wavefront

Each tile waits for its neighbors above and to the left. Events are named row by row. A cut is a left-justified staircase, and the last tile completes the wavefront.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	5	1
1	1	1	5	2
2	2	2	5	3
3	2	3	5	3
4	2	5	5	2
5	1	5	5	1
6	1	5	5	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–1	5	5
B	A	2–3	3	2
C	AB	3–5	1	2
D	A	2–4	2	1
E	ABD	4–5	2	3
F	$ABCDE$	6–6	5	5

A maximal concurrency witness The cut A enables $\{B, D\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 5, respectively. The cube counts include all choices of independently active events at every cut.

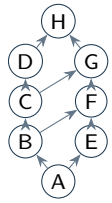
Dimension	Cubes	Alternating subtotal
0	10	10
1	12	-2
2	3	1

Among full schedules, 5 begin with A and 5 end with F. This count is combinatorial

and does not weight different execution times.

13.2 C02. 2 by 4 wavefront

Each tile waits for its neighbors above and to the left. Events are named row by row. A cut is a left-justified staircase, and the last tile completes the wavefront.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	14	1
1	1	1	14	2
2	2	2	14	3
3	2	3	14	4
4	3	6	14	4
5	2	9	14	3
6	2	14	14	2
7	1	14	14	1
8	1	14	14	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–1	14	14
B	A	2–3	9	5
C	AB	3–5	4	4
D	ABC	4–7	1	5
E	A	2–5	5	1
F	ABE	4–6	4	4
G	$ABCEF$	6–7	5	9
H	$ABCDEFG$	8–8	14	14

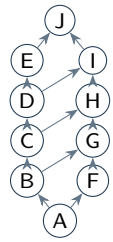
A maximal concurrency witness The cut A enables $\{B, E\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 14, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	15	15
1	20	-5
2	6	1

Among full schedules, 14 begin with A and 14 end with H. This count is combinatorial and does not weight different execution times.

13.3 C03. 2 by 5 wavefront

Each tile waits for its neighbors above and to the left. Events are named row by row. A cut is a left-justified staircase, and the last tile completes the wavefront.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	42	1
1	1	1	42	2
2	2	2	42	3
3	2	3	42	4
4	3	6	42	5
5	3	10	42	5
6	3	19	42	4
7	2	28	42	3
8	2	42	42	2
9	1	42	42	1
10	1	42	42	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–1	42	42
B	A	2–3	28	14
C	AB	3–5	14	10
D	ABC	4–7	5	10
E	$ABCD$	5–9	1	14
F	A	2–6	14	1
G	ABF	4–7	10	5
H	$ABCFG$	6–8	10	14
I	$ABCDFGH$	8–9	14	28
J	$ABCDEFGHI$	10–10	42	42

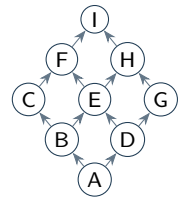
A maximal concurrency witness The cut A enables $\{B, F\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 42, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	21	21
1	30	-9
2	10	1

Among full schedules, 42 begin with A and 42 end with J. This count is combinatorial and does not weight different execution times.

13.4 C04. 3 by 3 wavefront

Each tile waits for its neighbors above and to the left. Events are named row by row. A cut is a left-justified staircase, and the last tile completes the wavefront.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	42	1
1	1	1	42	2
2	2	2	42	4
3	3	4	42	5
4	3	8	42	6
5	3	16	42	5
6	3	26	42	4
7	2	42	42	2
8	1	42	42	1
9	1	42	42	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–1	42	42
B	A	2–4	21	5
C	AB	3–7	5	5
D	A	2–4	21	5
E	ABD	4–6	12	12
F	$ABCDE$	6–8	5	21
G	AD	3–7	5	5
H	$ABDEG$	6–8	5	21
I	$ABCDEFGH$	9–9	42	42

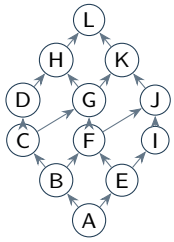
A maximal concurrency witness The cut ABD enables $\{C, E, G\}$. Its width is 3. Prefix and suffix schedule counts are 2 and 16, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	20	20
1	30	-10
2	12	2
3	1	1

Among full schedules, 42 begin with A and 42 end with I. This count is combinatorial and does not weight different execution times.

13.5 C05. 3 by 4 wavefront

Each tile waits for its neighbors above and to the left. Events are named row by row. A cut is a left-justified staircase, and the last tile completes the wavefront.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	462	1
1	1	1	462	2
2	2	2	462	4
3	3	4	462	6
4	4	9	462	8
5	4	20	462	9
6	5	45	462	9
7	4	91	462	8
8	4	182	462	6
9	3	294	462	4
10	2	462	462	2
11	1	462	462	1
12	1	462	462	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–1	462	462
B	A	2–4	252	42
C	AB	3–7	84	25
D	ABC	4–10	14	42
E	A	2–5	210	14
F	ABE	4–7	112	50
G	$ABCEF$	6–9	50	112
H	$ABCDEFG$	8–11	14	210
I	AE	3–9	42	14
J	$ABEFI$	6–10	25	84
K	$ABCEFGIJ$	9–11	42	252
L	$ABCDEFGHIJK$	12–12	462	462

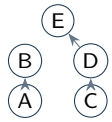
A maximal concurrency witness The cut ABE enables $\{C, F, I\}$. Its width is 3. Prefix and suffix schedule counts are 2 and 168, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	35	35
1	60	-25
2	30	5
3	4	1

Among full schedules, 462 begin with A and 462 end with L. This count is combinatorial and does not weight different execution times.

13.6 C06. Independent chains 2-3

Each chain is one worker's local sequence. There are no interworker messages, so a cut is a tuple of local prefix lengths. The number of full schedules is the multinomial coefficient for the chain lengths.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	10	2
1	2	2	10	4
2	3	4	10	5
3	3	7	10	4
4	2	10	10	2
5	1	10	10	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–4	4	1
B	A	2–5	1	4
C	$\varnothing$	1–3	6	1
D	C	2–4	3	3
E	CD	3–5	1	6

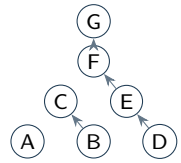
A maximal concurrency witness The cut $\varnothing$ enables $\{A, C\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 10, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	12	12
1	17	-5
2	6	1

Direct multinomial calculation gives 10 full schedules.

13.7 C07. Independent chains 1-2-4

Each chain is one worker's local sequence. There are no interworker messages, so a cut is a tuple of local prefix lengths. The number of full schedules is the multinomial coefficient for the chain lengths.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	105	3
1	3	3	105	8
2	5	8	105	12
3	6	19	105	13
4	6	39	105	12
5	5	70	105	8
6	3	105	105	3
7	1	105	105	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–7	15	15
B	∅	1–6	30	5
C	B	2–7	5	30
D	∅	1–4	60	3
E	D	2–5	30	12
F	DE	3–6	12	30
G	DEF	4–7	3	60

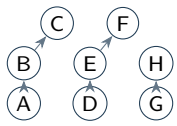
A maximal concurrency witness The cut ∅ enables $\{A, B, D\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 105, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	30	30
1	59	-29
2	38	9
3	8	1

Direct multinomial calculation gives 105 full schedules.

13.8 C08. Independent chains 3-3-2

Each chain is one worker's local sequence. There are no interworker messages, so a cut is a tuple of local prefix lengths. The number of full schedules is the multinomial coefficient for the chain lengths.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	560	3
1	3	3	560	9
2	6	9	560	17
3	9	26	560	23
4	10	70	560	23
5	9	170	560	17
6	6	350	560	9
7	3	560	560	3
8	1	560	560	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–6	210	10
B	A	2–7	60	60
C	AB	3–8	10	210
D	∅	1–6	210	10
E	D	2–7	60	60
F	DE	3–8	10	210
G	∅	1–7	140	20
H	G	2–8	20	140

A maximal concurrency witness The cut ∅ enables $\{A, D, G\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 560, respectively. The cube counts include all choices of independently active events at every cut.

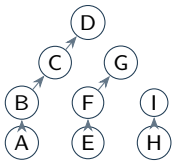
Dimension	Cubes	Alternating subtotal
0	48	48
1	104	-56
2	75	19
3	18	1

Direct multinomial calculation gives 560 full schedules.

13.9 C09. Independent chains 4-3-2

Each chain is one worker's local sequence. There are no interworker messages, so a

cut is a tuple of local prefix lengths. The number of full schedules is the multinomial coefficient for the chain lengths.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	1,260	3
1	3	3	1,260	9
2	6	9	1,260	17
3	9	26	1,260	24
4	11	71	1,260	27
5	11	180	1,260	24
6	9	410	1,260	17
7	6	805	1,260	9
8	3	1,260	1,260	3
9	1	1,260	1,260	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–6	560	10
B	A	2–7	210	60
C	AB	3–8	60	210
D	ABC	4–9	10	560
E	∅	1–7	420	15
F	E	2–8	105	105
G	EF	3–9	15	420
H	∅	1–8	280	35
I	H	2–9	35	280

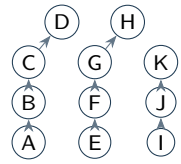
A maximal concurrency witness The cut ∅ enables $\{A, E, H\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 1,260, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	60	60
1	133	-73
2	98	25
3	24	1

Direct multinomial calculation gives 1,260 full schedules.

13.10 C10. Independent chains 4-4-3

Each chain is one worker's local sequence. There are no interworker messages, so a cut is a tuple of local prefix lengths. The number of full schedules is the multinomial coefficient for the chain lengths.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	11,550	3
1	3	3	11,550	9
2	6	9	11,550	18
3	10	27	11,550	29
4	14	80	11,550	38
5	16	230	11,550	41
6	16	630	11,550	38
7	14	1,610	11,550	29
8	10	3,710	11,550	18
9	6	7,350	11,550	9
10	3	11,550	11,550	3
11	1	11,550	11,550	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–8	4,200	35
B	*A*	2–9	1,260	280
C	*AB*	3–10	280	1,260
D	*ABC*	4–11	35	4,200
E	∅	1–8	4,200	35
F	*E*	2–9	1,260	280
G	*EF*	3–10	280	1,260
H	*EFG*	4–11	35	4,200
I	∅	1–9	3,150	70
J	*I*	2–10	630	630
K	*IJ*	3–11	70	3,150

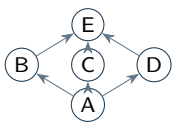
A maximal concurrency witness The cut ∅ enables $\{A, E, I\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 11,550, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	100	100
1	235	-135
2	184	49
3	48	1

Direct multinomial calculation gives 11,550 full schedules.

13.11 C11. Barriers 1-3-1

A barrier separates each pair of successive layers. Every event in the next layer depends on every event in the previous layer. Within one layer the events may execute in any order.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	6	1
1	1	1	6	3
2	3	3	6	6
3	3	6	6	3
4	1	6	6	1
5	1	6	6	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	6	6
B	*A*	2–4	2	2
C	*A*	2–4	2	2
D	*A*	2–4	2	2
E	*ABCD*	5–5	6	6

A maximal concurrency witness The cut *A* enables $\{B, C, D\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 6, respectively. The cube counts include all choices of independently active events at every cut.

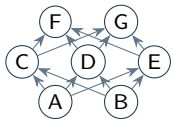
Dimension	Cubes	Alternating subtotal
0	10	10
1	14	-4
2	6	2
3	1	1

Among full schedules, 6 begin with A and 6 end with E. This count is combinatorial and does not weight different execution times.

13.12 C12. Barriers 2-3-2

A barrier separates each pair of successive layers. Every event in the next layer depends on every event in the previous layer.

Within one layer the events may execute in any order.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	24	2
1	2	2	24	2
2	1	2	24	3
3	3	6	24	6
4	3	12	24	3
5	1	12	24	2
6	2	24	24	2
7	1	24	24	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–2	12	12
B	∅	1–2	12	12
C	*AB*	3–5	8	8
D	*AB*	3–5	8	8
E	*AB*	3–5	8	8
F	*ABCDE*	6–7	12	12
G	*ABCDE*	6–7	12	12

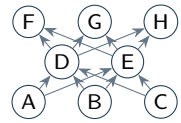
A maximal concurrency witness The cut AB enables $\{C, D, E\}$. Its width is 3. Prefix and suffix schedule counts are 2 and 12, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	14	14
1	20	-6
2	8	2
3	1	1

Among full schedules, 12 begin with A and 12 end with F. This count is combinatorial and does not weight different execution times.

13.13 C13. Barriers 3-2-3

A barrier separates each pair of successive layers. Every event in the next layer depends on every event in the previous layer. Within one layer the events may execute in any order.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	72	3
1	3	3	72	6
2	3	6	72	3
3	1	6	72	2
4	2	12	72	2
5	1	12	72	3
6	3	36	72	6
7	3	72	72	3
8	1	72	72	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–3	24	24
B	∅	1–3	24	24
C	∅	1–3	24	24
D	*ABC*	4–5	36	36
E	*ABC*	4–5	36	36
F	*ABCDE*	6–8	24	24
G	*ABCDE*	6–8	24	24
H	*ABCDE*	6–8	24	24

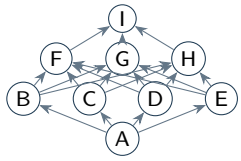
A maximal concurrency witness The cut ∅ enables $\{A, B, C\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 72, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	18	18
1	28	-10
2	13	3
3	2	1

Among full schedules, 24 begin with A and 24 end with F. This count is combinatorial and does not weight different execution times.

13.14 C14. Barriers 1-4-3-1

A barrier separates each pair of successive layers. Every event in the next layer depends on every event in the previous layer. Within one layer the events may execute in any order.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	144	1
1	1	1	144	4
2	4	4	144	12
3	6	12	144	12
4	4	24	144	4
5	1	24	144	3
6	3	72	144	6
7	3	144	144	3
8	1	144	144	1
9	1	144	144	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	144	144
B	*A*	2–5	36	36
C	*A*	2–5	36	36
D	*A*	2–5	36	36
E	*A*	2–5	36	36
F	*ABCDE*	6–8	48	48
G	*ABCDE*	6–8	48	48
H	*ABCDE*	6–8	48	48
I	*ABCDEFGH*	9–9	144	144

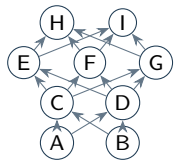
A maximal concurrency witness The cut *A* enables $\{B, C, D, E\}$. Its width is 4. Prefix and suffix schedule counts are 1 and 144, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	25	25
1	46	-21
2	30	9
3	9	0
4	1	1

Among full schedules, 144 begin with A and 144 end with I. This count is combinatorial and does not weight different execution times.

13.15 C15. Barriers 2-2-3-2

A barrier separates each pair of successive layers. Every event in the next layer depends on every event in the previous layer. Within one layer the events may execute in any order.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	48	2
1	2	2	48	2
2	1	2	48	2
3	2	4	48	2
4	1	4	48	3
5	3	12	48	6
6	3	24	48	3
7	1	24	48	2
8	2	48	48	2
9	1	48	48	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–2	24	24
B	∅	1–2	24	24
C	AB	3–4	24	24
D	AB	3–4	24	24
E	$ABCD$	5–7	16	16
F	$ABCD$	5–7	16	16
G	$ABCD$	5–7	16	16
H	$ABCDEFG$	8–9	24	24
I	$ABCDEFG$	8–9	24	24

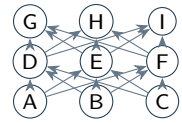
A maximal concurrency witness The cut $ABCD$ enables $\{E, F, G\}$. Its width is 3. Prefix and suffix schedule counts are 4 and 12, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	17	17
1	24	-7
2	9	2
3	1	1

Among full schedules, 24 begin with A and 24 end with H. This count is combinatorial and does not weight different execution times.

13.16 C16. Barriers 3-3-3

A barrier separates each pair of successive layers. Every event in the next layer depends on every event in the previous layer. Within one layer the events may execute in any order.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	216	3
1	3	3	216	6
2	3	6	216	3
3	1	6	216	3
4	3	18	216	6
5	3	36	216	3
6	1	36	216	3
7	3	108	216	6
8	3	216	216	3
9	1	216	216	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–3	72	72
B	∅	1–3	72	72
C	∅	1–3	72	72
D	*ABC*	4–6	72	72
E	*ABC*	4–6	72	72
F	*ABC*	4–6	72	72
G	*ABCDEF*	7–9	72	72
H	*ABCDEF*	7–9	72	72
I	*ABCDEF*	7–9	72	72

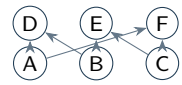
A maximal concurrency witness The cut ∅ enables $\{A, B, C\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 216, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	22	22
1	36	-14
2	18	4
3	3	1

Among full schedules, 72 begin with A and 72 end with G. This count is combinatorial and does not weight different execution times.

13.17 C17. Staggered relays 3-3

A receiving event depends on the indicated two neighboring positions when present in the previous layer. A receiving layer may start before its entire predecessor layer is done. The first enabling cut is shown in the event table.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	48	3
1	3	3	48	6
2	3	6	48	6
3	4	12	48	6
4	3	24	48	6
5	3	48	48	3
6	1	48	48	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–4	16	4
B	∅	1–4	16	4
C	∅	1–4	16	4
D	*AB*	3–6	4	16
E	*BC*	3–6	4	16
F	*AC*	3–6	4	16

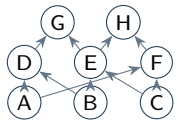
A maximal concurrency witness The cut ∅ enables $\{A, B, C\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 48, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	18	18
1	30	-12
2	15	3
3	2	1

Among full schedules, 16 begin with A and 16 end with D. This count is combinatorial and does not weight different execution times.

13.18 C18. Staggered relays 3-3-2

A receiving event depends on the indicated two neighboring positions when present in the previous layer. A receiving layer may start before its entire predecessor layer is done. The first enabling cut is shown in the event table.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	128	3
1	3	3	128	6
2	3	6	128	6
3	4	12	128	6
4	3	24	128	6
5	3	48	128	5
6	3	80	128	4
7	2	128	128	2
8	1	128	128	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–4	42	12
B	∅	1–4	43	10
C	∅	1–4	43	10
D	AB	3–7	10	16
E	BC	3–6	12	32
F	AC	3–7	10	16
G	$ABCDE$	6–8	16	64
H	$ABCEF$	6–8	16	64

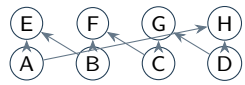
A maximal concurrency witness The cut ∅ enables $\{A, B, C\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 128, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	23	23
1	38	-15
2	18	3
3	2	1

Among full schedules, 43 begin with B and 64 end with G. This count is combinatorial and does not weight different execution times.

13.19 C19. Staggered relays 4-4

A receiving event depends on the indicated two neighboring positions when present in the previous layer. A receiving layer may start before its entire predecessor layer is done. The first enabling cut is shown in the event table.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	1,088	4
1	4	4	1,088	12
2	6	12	1,088	16
3	8	32	1,088	20
4	9	88	1,088	20
5	8	224	1,088	16
6	6	544	1,088	12
7	4	1,088	1,088	4
8	1	1,088	1,088	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–6	272	32
B	∅	1–6	272	32
C	∅	1–6	272	32
D	∅	1–6	272	32
E	AB	3–8	32	272
F	BC	3–8	32	272
G	CD	3–8	32	272
H	AD	3–8	32	272

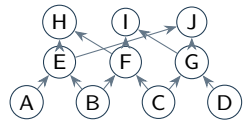
A maximal concurrency witness The cut ∅ enables $\{A, B, C, D\}$. Its width is 4. Prefix and suffix schedule counts are 1 and 1,088, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	47	47
1	104	-57
2	80	23
3	24	-1
4	2	1

Among full schedules, 272 begin with A and 272 end with E. This count is combinatorial and does not weight different execution times.

13.20 C20. Staggered relays 4-3-3

A receiving event depends on the indicated two neighboring positions when present in the previous layer. A receiving layer may start before its entire predecessor layer is done. The first enabling cut is shown in the event table.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	2,240	4
1	4	4	2,240	12
2	6	12	2,240	15
3	7	30	2,240	16
4	7	72	2,240	13
5	5	152	2,240	10
6	5	304	2,240	8
7	4	576	2,240	6
8	3	1,120	2,240	6
9	3	2,240	2,240	3
10	1	2,240	2,240	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–7	498	32
B	∅	1–5	622	128
C	∅	1–5	622	128
D	∅	1–7	498	32
E	AB	3–8	84	224
F	BC	3–8	104	160
G	CD	3–8	84	224
H	$ABCEF$	6–10	32	736
I	$BCDFG$	6–10	32	736
J	$ABCDEG$	7–10	160	768

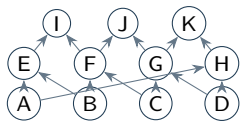
A maximal concurrency witness The cut ∅ enables $\{A, B, C, D\}$. Its width is 4. Prefix and suffix schedule counts are 1 and 2,240, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	46	46
1	93	-47
2	62	15
3	15	0
4	1	1

Among full schedules, 622 begin with B and 768 end with J. This count is combinatorial and does not weight different execution times.

13.21 C21. Staggered relays 4-4-3

A receiving event depends on the indicated two neighboring positions when present in the previous layer. A receiving layer may start before its entire predecessor layer is done. The first enabling cut is shown in the event table.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	12,672	4
1	4	4	12,672	12
2	6	12	12,672	16
3	8	32	12,672	20
4	9	88	12,672	20
5	8	224	12,672	19
6	9	592	12,672	18
7	7	1,424	12,672	16
8	7	3,392	12,672	13
9	5	7,104	12,672	8
10	3	12,672	12,672	3
11	1	12,672	12,672	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–7	3,060	96
B	∅	1–7	3,160	80
C	∅	1–6	3,292	256
D	∅	1–7	3,160	80
E	AB	3–10	336	768
F	BC	3–9	432	768
G	CD	3–9	432	768
H	AD	3–10	336	768
I	$ABCEF$	6–11	80	4,480
J	$BCDFG$	6–11	96	3,712
K	$ACDGH$	6–11	80	4,480

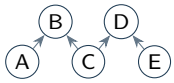
A maximal concurrency witness The cut ∅ enables $\{A, B, C, D\}$. Its width is 4. Prefix and suffix schedule counts are 1 and 12,672, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	68	68
1	149	-81
2	110	29
3	30	-1
4	2	1

Among full schedules, 3,292 begin with C and 4,480 end with I. This count is combinatorial and does not weight different execution times.

13.22 C22. 5-stage alternating fence

The alternating order represents requests at even positions and receipts at odd positions, with a receipt waiting for the adjacent requests that exist. Adjacent request dependencies overlap without forming a total barrier.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	16	3
1	3	3	16	6
2	3	6	16	5
3	3	10	16	4
4	2	16	16	2
5	1	16	16	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–4	5	2
B	AC	3–5	2	8
C	∅	1–3	6	4
D	CE	3–5	2	8
E	∅	1–4	5	2

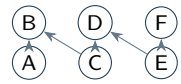
A maximal concurrency witness The cut ∅ enables $\{A, C, E\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 16, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	13	13
1	20	-7
2	9	2
3	1	1

Among full schedules, 6 begin with C and 8 end with B. This count is combinatorial and does not weight different execution times.

13.23 C23. 6-stage alternating fence

The alternating order represents requests at even positions and receipts at odd positions, with a receipt waiting for the adjacent requests that exist. Adjacent request dependencies overlap without forming a total barrier.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	61	3
1	3	3	61	7
2	4	7	61	9
3	5	16	61	9
4	4	33	61	7
5	3	61	61	3
6	1	61	61	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–5	16	5
B	AC	3–6	4	25
C	$\varnothing$	1–4	20	6
D	CE	3–6	6	20
E	$\varnothing$	1–4	25	4
F	E	2–6	5	16

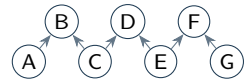
A maximal concurrency witness The cut $\varnothing$ enables $\{A, C, E\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 61, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	21	21
1	38	-17
2	22	5
3	4	1

Among full schedules, 25 begin with E and 25 end with B. This count is combinatorial and does not weight different execution times.

13.24 C24. 7-stage alternating fence

The alternating order represents requests at even positions and receipts at odd positions, with a receipt waiting for the adjacent requests that exist. Adjacent request dependencies overlap without forming a total barrier.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	272	4
1	4	4	272	12
2	6	12	272	15
3	7	30	272	16
4	7	72	272	13
5	5	152	272	8
6	3	272	272	3
7	1	272	272	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–6	61	16
B	AC	3–7	10	96
C	$\varnothing$	1–5	75	16
D	CE	3–7	12	80
E	$\varnothing$	1–5	75	16
F	EG	3–7	10	96
G	$\varnothing$	1–6	61	16

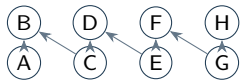
A maximal concurrency witness The cut $\varnothing$ enables $\{A, C, E, G\}$. Its width is 4. Prefix and suffix schedule counts are 1 and 272, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	34	34
1	71	-37
2	51	14
3	14	0
4	1	1

Among full schedules, 75 begin with C and 96 end with B. This count is combinatorial and does not weight different execution times.

13.25 C25. 8-stage alternating fence

The alternating order represents requests at even positions and receipts at odd positions, with a receipt waiting for the adjacent requests that exist. Adjacent request dependencies overlap without forming a total barrier.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	1,385	4
1	4	4	1,385	13
2	7	13	1,385	21
3	10	39	1,385	27
4	11	113	1,385	27
5	10	302	1,385	21
6	7	723	1,385	13
7	4	1,385	1,385	4
8	1	1,385	1,385	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–7	272	61
B	AC	3–8	32	427
C	$\varnothing$	1–6	336	50
D	CE	3–8	40	350
E	$\varnothing$	1–6	350	40
F	EG	3–8	50	336
G	$\varnothing$	1–6	427	32
H	G	2–8	61	272

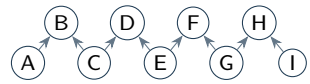
A maximal concurrency witness The cut $\varnothing$ enables $\{A, C, E, G\}$. Its width is 4. Prefix and suffix schedule counts are 1 and 1,385, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	55	55
1	130	-75
2	111	36
3	40	-4
4	5	1

Among full schedules, 427 begin with G and 427 end with B. This count is combinatorial and does not weight different execution times.

13.26 C26. 9-stage alternating fence

The alternating order represents requests at even positions and receipts at odd positions, with a receipt waiting for the adjacent requests that exist. Adjacent request dependencies overlap without forming a total barrier.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	7,936	5
1	5	5	7,936	20
2	10	20	7,936	34
3	14	68	7,936	44
4	17	216	7,936	47
5	16	648	7,936	40
6	13	1,776	7,936	27
7	8	4,240	7,936	14
8	4	7,936	7,936	4
9	1	7,936	7,936	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–8	1,385	272
B	AC	3–9	122	2,176
C	∅	1–7	1,708	192
D	CE	3–9	150	1,792
E	∅	1–7	1,750	160
F	EG	3–9	150	1,792
G	∅	1–7	1,708	192
H	GI	3–9	122	2,176
I	∅	1–8	1,385	272

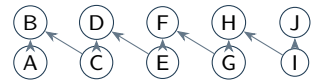
A maximal concurrency witness The cut ∅ enables $\{A, C, E, G, I\}$. Its width is 5. Prefix and suffix schedule counts are 1 and 7,936, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	89	89
1	235	-146
2	233	87
3	105	-18
4	20	2
5	1	1

Among full schedules, 1,750 begin with E and 2,176 end with B. This count is combinatorial and does not weight different execution times.

13.27 C27. 10-stage alternating fence

The alternating order represents requests at even positions and receipts at odd positions, with a receipt waiting for the adjacent requests that exist. Adjacent request dependencies overlap without forming a total barrier.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	50,521	5
1	5	5	50,521	21
2	11	21	50,521	42
3	18	80	50,521	64
4	24	293	50,521	78
5	26	1,023	50,521	78
6	24	3,367	50,521	64
7	18	10,036	50,521	42
8	11	25,953	50,521	21
9	5	50,521	50,521	5
10	1	50,521	50,521	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–9	7,936	1,385
B	*AC*	3–10	544	12,465
C	∅	1–8	9,792	854
D	*CE*	3–10	672	10,248
E	∅	1–8	10,080	700
F	*EG*	3–10	700	10,080
G	∅	1–8	10,248	672
H	*GI*	3–10	854	9,792
I	∅	1–8	12,465	544
J	*I*	2–10	1,385	7,936

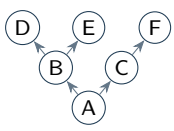
A maximal concurrency witness The cut ∅ enables $\{A, C, E, G, I\}$. Its width is 5. Prefix and suffix schedule counts are 1 and 50,521, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	144	144
1	420	-276
2	474	198
3	256	-58
4	65	7
5	6	1

Among full schedules, 12,465 begin with I and 12,465 end with B. This count is combinatorial and does not weight different execution times.

13.28 C28. 6-event binary dissemination

A root notification enables its two child notifications, with the final layer possibly incomplete. Each edge represents a forward dependency. Reversing every edge models gathering with the same number of full schedules and the complementary position profile.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	20	1
1	1	1	20	2
2	2	2	20	5
3	4	5	20	8
4	4	11	20	7
5	3	20	20	3
6	1	20	20	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	20	20
B	*A*	2–4	12	2
C	*A*	2–5	8	2
D	*AB*	3–6	3	6
E	*AB*	3–6	3	6
F	*AC*	3–6	2	8

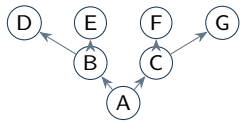
A maximal concurrency witness The cut AB enables $\{C, D, E\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 12, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	16	16
1	26	-10
2	13	3
3	2	1

Among full schedules, 20 begin with A and 8 end with F. This count is combinatorial and does not weight different execution times.

13.29 C29. 7-event binary dissemination

A root notification enables its two child notifications, with the final layer possibly incomplete. Each edge represents a forward dependency. Reversing every edge models gathering with the same number of full schedules and the complementary position profile.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	80	1
1	1	1	80	2
2	2	2	80	6
3	5	6	80	12
4	6	16	80	14
5	6	40	80	12
6	4	80	80	4
7	1	80	80	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	80	80
B	A	2–5	40	4
C	A	2–5	40	4
D	AB	3–7	8	20
E	AB	3–7	8	20
F	AC	3–7	8	20
G	AC	3–7	8	20

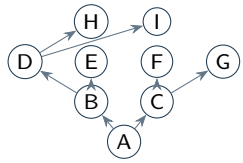
A maximal concurrency witness The cut ABC enables $\{D, E, F, G\}$. Its width is 4. Prefix and suffix schedule counts are 2 and 24, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	26	26
1	51	-25
2	35	10
3	10	0
4	1	1

Among full schedules, 80 begin with A and 20 end with D. This count is combinatorial and does not weight different execution times.

13.30 C30. 9-event binary dissemination

A root notification enables its two child notifications, with the final layer possibly incomplete. Each edge represents a forward dependency. Reversing every edge models gathering with the same number of full schedules and the complementary position profile.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	896	1
1	1	1	896	2
2	2	2	896	6
3	5	6	896	14
4	8	18	896	24
5	11	56	896	32
6	12	168	896	32
7	10	448	896	20
8	5	896	896	5
9	1	896	896	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	896	896
B	*A*	2–5	560	16
C	*A*	2–7	336	16
D	*AB*	3–7	240	40
E	*AB*	3–9	80	140
F	*AC*	3–9	48	168
G	*AC*	3–9	48	168
H	*ABD*	4–9	40	210
I	*ABD*	4–9	40	210

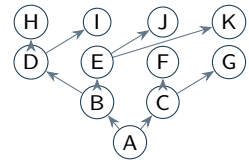
A maximal concurrency witness The cut $ABCD$ enables $\{E, F, G, H, I\}$. Its width is 5. Prefix and suffix schedule counts are 3 and 120, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	56	56
1	136	-80
2	126	46
3	56	-10
4	12	2
5	1	1

Among full schedules, 896 begin with A and 210 end with H. This count is combinatorial and does not weight different execution times.

13.31 C31. 11-event binary dissemination

A root notification enables its two child notifications, with the final layer possibly incomplete. Each edge represents a forward dependency. Reversing every edge models gathering with the same number of full schedules and the complementary position profile.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	19,200	1
1	1	1	19,200	2
2	2	2	19,200	6
3	5	6	19,200	16
4	10	20	19,200	34
5	16	72	19,200	60
6	24	280	19,200	84
7	27	1,040	19,200	84
8	23	3,440	19,200	63
9	15	9,600	19,200	30
10	6	19,200	19,200	6
11	1	19,200	19,200	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	19,200	19,200
B	*A*	2–5	13,440	160
C	*A*	2–9	5,760	160
D	*AB*	3–9	4,480	280
E	*AB*	3–9	4,480	280
F	*AC*	3–11	640	2,880
G	*AC*	3–11	640	2,880
H	*ABD*	4–11	560	3,360
I	*ABD*	4–11	560	3,360
J	*ABE*	4–11	560	3,360
K	*ABE*	4–11	560	3,360

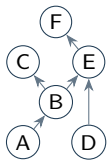
A maximal concurrency witness The cut $ABCDE$ enables $\{F, G, H, I, J, K\}$. Its width is 6. Prefix and suffix schedule counts are 8 and 720, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	131	131
1	386	-255
2	456	201
3	276	-75
4	90	15
5	15	0
6	1	1

Among full schedules, 19,200 begin with A and 3,360 end with H. This count is combinatorial and does not weight different execution times.

13.32 C32. One cross-message

Two workers have fixed local orders and exchange messages with the additional causal dependencies shown in the diagram. A cut must pass both the local prefix and cross-message checks.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	10	2
1	2	2	10	3
2	2	3	10	3
3	2	4	10	3
4	2	7	10	3
5	2	10	10	2
6	1	10	10	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–2	7	3
B	*A*	2–3	4	6
C	*AB*	3–6	1	3
D	∅	1–4	3	1
E	*ABD*	4–5	6	4
F	*ABDE*	5–6	3	7

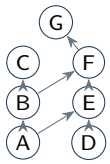
A maximal concurrency witness The cut ∅ enables $\{A, D\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 10, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	12	12
1	16	-4
2	5	1

Among full schedules, 7 begin with A and 7 end with F. This count is combinatorial and does not weight different execution times.

13.33 C33. Two cross-messages

Two workers have fixed local orders and exchange messages with the additional causal dependencies shown in the diagram. A cut must pass both the local prefix and cross-message checks.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	19	2
1	2	2	19	3
2	2	3	19	4
3	3	6	19	4
4	2	9	19	3
5	2	14	19	3
6	2	19	19	2
7	1	19	19	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–2	12	7
B	A	2–4	5	6
C	AB	3–7	1	5
D	$\varnothing$	1–4	7	1
E	AD	3–5	6	4
F	$ABDE$	5–6	10	9
G	$ABDEF$	6–7	5	14

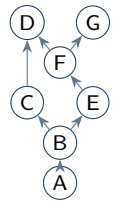
A maximal concurrency witness The cut $\varnothing$ enables $\{A, D\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 19, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	15	15
1	21	-6
2	7	1

Among full schedules, 12 begin with A and 14 end with G. This count is combinatorial and does not weight different execution times.

13.34 C34. Round trip

Two workers have fixed local orders and exchange messages with the additional causal dependencies shown in the diagram. A cut must pass both the local prefix and cross-message checks.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	7	1
1	1	1	7	1
2	1	1	7	2
3	2	2	7	3
4	2	3	7	3
5	2	4	7	3
6	2	7	7	2
7	1	7	7	0

Event placement The prerequisite column includes all strict predecessors. Endpoint

counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	7	7
B	*A*	2–2	7	7
C	*AB*	3–6	2	1
D	*ABCEF*	6–7	3	4
E	*AB*	3–4	5	2
F	*ABE*	4–5	3	4
G	*ABEF*	5–7	1	3

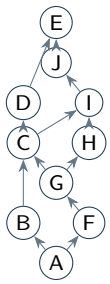
A maximal concurrency witness The cut AB enables $\{C, E\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 7, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	12	12
1	15	-3
2	4	1

Among full schedules, 7 begin with A and 4 end with D. This count is combinatorial and does not weight different execution times.

13.35 C35. Two round trips

Two workers have fixed local orders and exchange messages with the additional causal dependencies shown in the diagram. A cut must pass both the local prefix and cross-message checks.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	24	1
1	1	1	24	2
2	2	2	24	3
3	2	3	24	3
4	2	4	24	3
5	2	7	24	3
6	2	10	24	3
7	2	17	24	3
8	2	24	24	2
9	1	24	24	1
10	1	24	24	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–1	24	24
B	*A*	2–5	7	3
C	*ABFG*	5–6	12	12
D	*ABCFG*	6–9	3	7
E	*ABCDFGHIJ*	10–10	24	24
F	*A*	2–3	17	7
G	*AF*	3–4	10	14
H	*AFG*	4–7	3	3
I	*ABCFGH*	7–8	14	10
J	*ABCFGHI*	8–9	7	17

A maximal concurrency witness The cut A enables $\{B, F\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 24, respectively. The cube counts include all choices of independently active events at every cut.

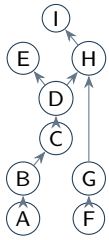
Dimension	Cubes	Alternating subtotal
0	18	18
1	24	-6
2	7	1

Among full schedules, 24 begin with A and 24 end with E. This count is com-

binatorial and does not weight different execution times.

13.36 C36. Late notification

Two workers have fixed local orders and exchange messages with the additional causal dependencies shown in the diagram. A cut must pass both the local prefix and cross-message checks.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	51	2
1	2	2	51	4
2	3	4	51	5
3	3	7	51	5
4	3	11	51	5
5	3	16	51	4
6	2	21	51	3
7	2	36	51	3
8	2	51	51	2
9	1	51	51	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	$\varnothing$	1–3	35	3
B	A	2–4	22	9
C	AB	3–5	12	18
D	ABC	4–6	5	30
E	$ABCD$	5–9	1	15
F	$\varnothing$	1–6	16	1
G	F	2–7	3	6
H	$ABCDFG$	7–8	30	21
I	$ABCDFGH$	8–9	15	36

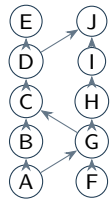
A maximal concurrency witness The cut $\varnothing$ enables $\{A, F\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 51, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	22	22
1	33	-11
2	12	1

Among full schedules, 35 begin with A and 36 end with I. This count is combinatorial and does not weight different execution times.

13.37 C37. Pipeline backpressure

Two workers have fixed local orders and exchange messages with the additional causal dependencies shown in the diagram. A cut must pass both the local prefix and cross-message checks.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	98	2
1	2	2	98	3
2	2	3	98	3
3	2	5	98	3
4	2	7	98	4
5	3	14	98	5
6	3	26	98	5
7	3	43	98	4
8	2	60	98	3
9	2	98	98	2
10	1	98	98	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–2	57	41
B	A	2–6	16	4
C	$ABFG$	5–7	45	18
D	$ABCFG$	6–8	20	42
E	$ABCDFG$	7–10	5	38
F	∅	1–3	41	16
G	AF	3–4	50	48
H	AFG	4–8	18	5
I	$AFGH$	5–9	4	22
J	$ABCDFGHI$	9–10	38	60

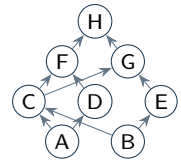
A maximal concurrency witness The cut ∅ enables $\{A, F\}$. Its width is 2. Prefix and suffix schedule counts are 1 and 98, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	23	23
1	34	-11
2	12	1

Among full schedules, 57 begin with A and 60 end with J. This count is combinatorial and does not weight different execution times.

13.38 C38. Butterfly split and merge

A shared middle event feeds two distinct completion branches.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	42	2
1	2	2	42	4
2	3	4	42	5
3	3	8	42	6
4	3	16	42	5
5	3	26	42	4
6	2	42	42	2
7	1	42	42	1
8	1	42	42	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–3	21	5
B	∅	1–3	21	5
C	AB	3–5	12	12
D	A	2–6	5	5
E	B	2–6	5	5
F	$ABCD$	5–7	5	21
G	$ABCE$	5–7	5	21
H	$ABCDEFG$	8–8	42	42

A maximal concurrency witness The cut AB enables $\{C, D, E\}$. Its width is 3. Prefix and suffix schedule counts are 2 and 16,

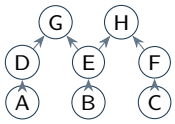
respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	19	19
1	29	-10
2	12	2
3	1	1

Among full schedules, 21 begin with A and 42 end with H. This count is combinatorial and does not weight different execution times.

13.39 C39. Independent quorum responses

Two later receipts depend on partly overlapping responder sets.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	252	3
1	3	3	252	9
2	6	9	252	15
3	7	24	252	15
4	6	54	252	11
5	5	102	252	7
6	3	162	252	4
7	2	252	252	2
8	1	252	252	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–6	81	6
B	∅	1–5	90	12
C	∅	1–6	81	6
D	A	2–7	16	36
E	B	2–6	20	60
F	C	2–7	16	36
G	$ABDE$	5–8	6	126
H	$BCEF$	5–8	6	126

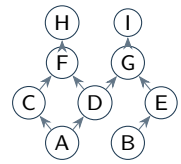
A maximal concurrency witness The cut ∅ enables $\{A, B, C\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 252, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	34	34
1	66	-32
2	41	9
3	8	1

Among full schedules, 90 begin with B and 126 end with G. This count is combinatorial and does not weight different execution times.

13.40 C40. Speculative read frontier

Two possible read certificates advance through partially shared prerequisites.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	186	2
1	2	2	186	5
2	4	5	186	8
3	4	11	186	8
4	4	22	186	7
5	4	38	186	7
6	4	74	186	6
7	3	130	186	4
8	2	186	186	2
9	1	186	186	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–3	118	16
B	∅	1–6	68	2
C	*A*	2–7	31	6
D	*A*	2–5	35	36
E	*B*	2–7	16	12
F	*ACD*	4–8	10	32
G	*ABDE*	5–8	24	42
H	*ACDF*	5–9	2	88
I	*ABDEG*	6–9	6	98

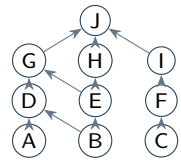
A maximal concurrency witness The cut A enables $\{B, C, D\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 118, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	29	29
1	49	-20
2	23	3
3	2	1

Among full schedules, 118 begin with A and 98 end with I. This count is combinatorial and does not weight different execution times.

13.41 C41. Recovery gates

Three local recoveries gate one joint output. Finishing one recovery never authorizes the output before the other paths complete.



Covering dependencies, directed from prerequisite to dependent event.

Rank census p counts paths into a cut and q counts paths out. The last column sums enabled choices over the cuts in the row.

k	Cuts	$\sum p$	$\sum pq$	Enabled
0	1	1	1,176	3
1	3	3	1,176	8
2	5	8	1,176	14
3	8	22	1,176	20
4	9	59	1,176	21
5	9	149	1,176	20
6	8	353	1,176	14
7	5	707	1,176	8
8	3	1,176	1,176	3
9	1	1,176	1,176	1
10	1	1,176	1,176	0

Event placement The prerequisite column includes all strict predecessors. Endpoint counts show how many full schedules put the event at the earliest or latest position.

Event	Needs	Range	First	Last
A	∅	1–7	280	20
B	∅	1–5	504	20
C	∅	1–7	392	14
D	*AB*	3–8	80	140
E	*B*	2–7	140	80
F	*C*	2–8	98	98
G	*ABDE*	5–9	20	504
H	*BE*	3–9	20	280
I	*CF*	3–9	14	392
J	*ABCDEFGHI*	10–10	1,176	1,176

A maximal concurrency witness The cut $\varnothing$ enables $\{A, B, C\}$. Its width is 3. Prefix and suffix schedule counts are 1 and 1,176, respectively. The cube counts include all choices of independently active events at every cut.

Dimension	Cubes	Alternating subtotal
0	53	53
1	112	-59
2	78	19
3	18	1

Among full schedules, 504 begin with B and 1,176 end with J. This count is combinatorial and does not weight different execution times.

14 Atlas of quorum availability

Quorums are described by their minimal certificates. The first and second families can be used for two phases of a ballot, or as distinct observation requirements. A blocking set hits all certificates in its family. See Theorem 9.1. Examples list short certificate samples when a full family would be long. The source bundle contains every set.

14.1 Q01. 3 and 3 of 5

Any 3 nodes authorize the first phase and any 3 nodes authorize the second. The threshold sum is 6; the comparison is with 5, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Phase 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	10	10	10	10

Cross-certificate test The pair ABC and ADE meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	30	3/10
2	60	3/5
3	10	1/10

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	10	0	0	0
3	0	0	0	10
4	0	0	0	5
5	0	0	0	1

One correlated failure Removing ABC leaves DE. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 6 cases. For phase 2, A is pivotal in 6 cases. If all

survivor sets are equally likely, the probability of both certificates being possible is 1/2.

14.2 Q02. 2 and 4 of 5

Any 2 nodes authorize the first phase and any 4 nodes authorize the second. The threshold sum is 6; the comparison is with 5, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, BD, CD, AE, and 3 more
Phase 2	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$
Blocks 1	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$
Blocks 2	AB, AC, BC, AD, BD, CD, AE, and 3 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	10	0	0	10
4	0	5	5	0

Cross-certificate test The pair AB and $ACDE$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	20	2/5
2	30	3/5

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	0	10	0	0
3	0	10	0	0
4	0	0	0	5
5	0	0	0	1

One correlated failure Removing ABC leaves DE. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 4 cases. For phase 2, A is pivotal in 4 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 3/16.

14.3 Q03. 2 and 3 of 5

Any 2 nodes authorize the first phase and any 3 nodes authorize the second. The threshold sum is 5; the comparison is with 5, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, BD, CD, AE, and 3 more
Phase 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 1	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	10	0	0	0
3	0	10	0	10
4	0	0	5	0

Cross-certificate test The pair AB and CDE meets in $\varnothing$. The smallest intersection has size 0. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
0	10	1/10
1	60	3/5
2	30	3/10

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	0	10	0	0
3	0	0	0	10
4	0	0	0	5
5	0	0	0	1

One correlated failure Removing ABC leaves DE. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 4 cases. For phase 2, A is pivotal in 6 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/2.

14.4 Q04. 2 and 5 of 6

Any 2 nodes authorize the first phase and any 5 nodes authorize the second. The threshold sum is 7; the comparison is with 6, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, BD, CD, AE, and 8 more
Phase 2	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$
Blocks 1	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$
Blocks 2	AB, AC, BC, AD, BD, CD, AE, and 8 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	15	0	0	15
5	0	6	6	0

Cross-certificate test The pair AB and $ACDEF$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	30	1/3
2	60	2/3

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	0	15	0	0
3	0	20	0	0
4	0	15	0	0
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing ABC leaves DEF. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 4 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 5 cases. For phase 2, A is pivotal in 5 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 7/64.

14.5 Q05. 3 and 4 of 6

Any 3 nodes authorize the first phase and any 4 nodes authorize the second. The threshold sum is 7; the comparison is with 6, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 13 more
Phase 2	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$, $ABCF$, $ABDF$, and 8 more
Blocks 1	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$, $ABCF$, $ABDF$, and 8 more
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 13 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	20	0	0	20
4	0	15	15	0

Cross-certificate test The pair ABC and $ADEF$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	60	1/5
2	180	3/5
3	60	1/5

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	15	0	0	0
3	0	20	0	0
4	0	0	0	15
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing ABC leaves DEF. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 10 cases. For phase 2, A is pivotal in 10 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 11/32.

14.6 Q06. 4 and 4 of 7

Any 4 nodes authorize the first phase and any 4 nodes authorize the second. The threshold sum is 8; the comparison is with 7, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$, $ABCF$, $ABDF$, and 28 more
Phase 2	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$, $ABCF$, $ABDF$, and 28 more
Blocks 1	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$, $ABCF$, $ABDF$, and 28 more
Blocks 2	$ABCD$, $ABCE$, $ABDE$, $ACDE$, $BCDE$, $ABCF$, $ABDF$, and 28 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
4	35	35	35	35

Cross-certificate test The pair $ABCD$ and $AEFG$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	140	4/35
2	630	18/35
3	420	12/35
4	35	1/35

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	21	0	0	0
3	35	0	0	0
4	0	0	0	35
5	0	0	0	21
6	0	0	0	7
7	0	0	0	1

One correlated failure Removing ABC leaves $DEFG$. Availability is yes for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 3 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 20 cases. For phase 2, A is pivotal in 20 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/2.

14.7 Q07. 3 and 5 of 7

Any 3 nodes authorize the first phase and any 5 nodes authorize the second. The threshold sum is 8; the comparison is with 7, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 28 more
Phase 2	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$, $ABCDG$, and 14 more
Blocks 1	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$, $ABCDG$, and 14 more
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 28 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	35	0	0	35
5	0	21	21	0

Cross-certificate test The pair ABC and $ADEFG$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	105	1/7
2	420	4/7
3	210	2/7

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	21	0	0	0
3	0	35	0	0
4	0	35	0	0
5	0	0	0	21
6	0	0	0	7
7	0	0	0	1

One correlated failure Removing ABC leaves $DEFG$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 4 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 15 cases. For phase 2, A is pivotal in 15 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 29/128.

14.8 Q08. 2 and 6 of 7

Any 2 nodes authorize the first phase and any 6 nodes authorize the second. The threshold sum is 8; the comparison is with 7, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, BD, CD, AE, and 14 more
Phase 2	$ABCDEF$, $ABCDEG$, $ABCDFG$, $ABCEFG$, $ABDEFG$, $ACDEFG$, $BCDEFG$
Blocks 1	$ABCDEF$, $ABCDEG$, $ABCDFG$, $ABCEFG$, $ABDEFG$, $ACDEFG$, $BCDEFG$
Blocks 2	AB, AC, BC, AD, BD, CD, AE, and 14 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	21	0	0	21
6	0	7	7	0

Cross-certificate test The pair AB and $ACDEFG$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	42	2/7
2	105	5/7

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	0	21	0	0
3	0	35	0	0
4	0	35	0	0
5	0	21	0	0
6	0	0	0	7
7	0	0	0	1

One correlated failure Removing ABC leaves $DEFG$. Availability is yes for phase 1 and no for phase 2. Over all failure sets

the sharp uniform tolerances are 5 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 6 cases. For phase 2, A is pivotal in 6 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/16.

14.9 Q09. 3 and 6 of 8

Any 3 nodes authorize the first phase and any 6 nodes authorize the second. The threshold sum is 9; the comparison is with 8, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 49 more
Phase 2	$ABCDEF$, $ABCDEG$, $ABCDFG$, $ABCEFG$, $ABDEFG$, $ACDEFG$, $BCDEFG$, and 21 more
Blocks 1	$ABCDEF$, $ABCDEG$, $ABCDFG$, $ABCEFG$, $ABDEFG$, $ACDEFG$, $BCDEFG$, and 21 more
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 49 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	56	0	0	56
6	0	28	28	0

Cross-certificate test The pair ABC and $ADEFGH$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	168	3/28
2	840	15/28
3	560	5/14

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	28	0	0	0
3	0	56	0	0
4	0	70	0	0
5	0	56	0	0
6	0	0	0	28
7	0	0	0	8
8	0	0	0	1

One correlated failure Removing ABC leaves $DEFGH$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 5 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 21 cases. For phase 2, A is pivotal in 21 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 37/256.

14.10 Q10. 5 and 5 of 9

Any 5 nodes authorize the first phase and any 5 nodes authorize the second. The threshold sum is 10; the comparison is with 9, the number of nodes.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$, $ABCDG$, and 119 more
Phase 2	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$, $ABCDG$, and 119 more
Blocks 1	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$, $ABCDG$, and 119 more
Blocks 2	$ABCDE$, $ABCDF$, $ABCEF$, $ABDEF$, $ACDEF$, $BCDEF$, $ABCDG$, and 119 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
5	126	126	126	126

Cross-certificate test The pair $ABCDE$ and $AFGHI$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	630	5/126
2	5,040	20/63
3	7,560	10/21
4	2,520	10/63
5	126	1/126

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	9	0	0	0
2	36	0	0	0
3	84	0	0	0
4	126	0	0	0
5	0	0	0	126
6	0	0	0	84
7	0	0	0	36
8	0	0	0	9
9	0	0	0	1

One correlated failure Removing ABC leaves $DEFGHI$. Availability is yes for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 4 and 4 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 70 cases. For phase 2, A is pivotal in 70 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/2.

14.11 Q11. Weighted thresholds 4 and 6 over 5 nodes

Node weights in alphabetic order are $(3, 2, 2, 1, 1)$, with total 9. The two weight thresholds are 4 and 6. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, AE, BDE, CDE
Phase 2	ABC, ABD, ACD, ABE, ACE, $BCDE$
Blocks 1	ABC, ABD, ACD, ABE, ACE, $BCDE$
Blocks 2	AB, AC, BC, AD, AE, BDE, CDE

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	5	0	0	5
3	2	5	5	2
4	0	1	1	0

Cross-certificate test The pair AB and ACD meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	24	4/7
2	16	8/21
3	2	1/21

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	5	5	0	0
3	0	5	0	5
4	0	0	0	5
5	0	0	0	1

One correlated failure Removing AB leaves CDE. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 9 cases. For phase 2, A is pivotal in 9 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 11/32.

14.12 Q12. Weighted thresholds 5 and 5 over 5 nodes

Node weights in alphabetic order are $(5, 1, 1, 1, 1)$, with total 9. The two weight thresholds are 5 and 5. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	A
Phase 2	A
Blocks 1	A
Blocks 2	A

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	1	1	1	1

Cross-certificate test The pair A and A meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	1	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	4	0	0	1
2	6	0	0	4
3	4	0	0	6
4	1	0	0	4
5	0	0	0	1

One correlated failure Removing AB leaves CDE. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 0 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 16 cases. For phase 2, A is pivotal in 16 cases. If all survivor sets are equally likely, the probability of both certificates being possible is $1/2$.

14.13 Q13. Weighted thresholds 4 and 5 over 5 nodes

Node weights in alphabetic order are $(5, 1, 1, 1, 1)$, with total 9. The two weight thresholds are 4 and 5. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	A, $BCDE$
Phase 2	A
Blocks 1	AB, AC, AD, AE
Blocks 2	A

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	1	1	0	1
2	0	0	4	0
4	1	0	0	0

Cross-certificate test The pair $BCDE$ and A meets in $\varnothing$. The smallest intersection has size 0. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
0	1	1/2
1	1	1/2

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	4	0	0	1
2	6	0	0	4
3	4	0	0	6
4	0	1	0	4
5	0	0	0	1

One correlated failure Removing AB leaves CDE. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 15 cases. For phase 2, A is pivotal in 16 cases. If all survivor sets are equally likely, the probability of both certificates being possible is $1/2$.

14.14 Q14. Weighted thresholds 5 and 5 over 5 nodes

Node weights in alphabetic order are $(2, 2, 2, 2, 2)$, with total 10. The two weight thresholds are 5 and 5. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The

following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Phase 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	10	10	10	10

Cross-certificate test The pair ABC and ADE meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	30	3/10
2	60	3/5
3	10	1/10

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	10	0	0	0
3	0	0	0	10
4	0	0	0	5
5	0	0	0	1

One correlated failure Removing AB leaves CDE. Availability is yes for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 6 cases. For phase 2, A is pivotal in 6 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/2.

14.15 Q15. Weighted thresholds 5 and 8 over 6 nodes

Node weights in alphabetic order are $(4, 3, 2, 1, 1, 1)$, with total 12. The two weight thresholds are 5 and 8. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, AE, AF, BDE, and 3 more
Phase 2	ABC, ABD, ABE, ABF, $ACDE$, $ACDF$, $ACEF$, and 1 more
Blocks 1	ABC, ABD, ABE, ABF, $ACDE$, $ACDF$, $ACEF$, and 1 more
Blocks 2	AB, AC, BC, AD, AE, AF, BDE, and 3 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	6	0	0	6
3	3	4	4	3
4	1	3	3	1
5	0	1	1	0

Cross-certificate test The pair AB and $ACDE$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	45	9/16
2	28	7/20
3	6	3/40
4	1	1/80

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	9	6	0	0
3	4	12	0	4
4	0	6	0	9
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing AB leaves $CDEF$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 18 cases. For phase 2, A is pivotal in 18 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 5/16.

14.16 Q16. Weighted thresholds 6 and 7 over 6 nodes

Node weights in alphabetic order are $(3, 3, 3, 1, 1, 1)$, with total 12. The two weight thresholds are 6 and 7. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, $ADEF$, $BDEF$, $CDEF$
Phase 2	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 1	ABC, ABD, ACD, BCD, ABE, ACE, BCE, and 3 more
Blocks 2	AB, AC, BC, $ADEF$, $BDEF$, $CDEF$

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	3	0	0	3
3	0	10	10	0
4	3	0	0	3

Cross-certificate test The pair AB and ACD meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	30	1/2
2	30	1/2

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	12	3	0	0
3	10	0	0	10
4	0	3	0	12
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing AB leaves $CDEF$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 15 cases. For phase 2, A is pivotal in 15 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 29/64.

14.17 Q17. Weighted thresholds 6 and 10 over 7 nodes

Node weights in alphabetic order are $(4, 4, 2, 2, 1, 1, 1)$, with total 15. The two weight thresholds are 6 and 10. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BC, AD, BD, AEF, BEF, and 7 more
Phase 2	ABC, ABD, $ABEF$, $ABEG$, $ABFG$, $ACDEF$, $BCDEF$, and 4 more
Blocks 1	ABC, ABD, $ABEF$, $ABEG$, $ABFG$, $ACDEF$, $BCDEF$, and 4 more
Blocks 2	AB, AC, BC, AD, BD, AEF, BEF, and 7 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	5	0	0	5
3	6	2	2	6
4	3	3	3	3
5	0	6	6	0

Cross-certificate test The pair AB and $ACDEF$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	70	5/11
2	54	27/77
3	24	12/77
4	6	3/77

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	16	5	0	0
3	10	23	0	2
4	2	23	0	10
5	0	5	0	16
6	0	0	0	7
7	0	0	0	1

One correlated failure Removing AB leaves $CDEFG$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 28 cases. For phase 2, A is pivotal in 28 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 9/32.

14.18 Q18. Weighted thresholds 8 and 10 over 7 nodes

Node weights in alphabetic order are $(5, 3, 3, 2, 2, 1, 1)$, with total 17. The two weight thresholds are 8 and 10. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets.

Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, AC, BCD, BCE, ADE, ADF, AEF, and 7 more
Phase 2	ABC, ABD, ACD, ABE, ACE, $BCDE$, $ADEF$, and 5 more
Blocks 1	ABC, ABD, ACD, ABE, ACE, $BCDE$, $ADEF$, and 5 more
Blocks 2	AB, AC, BCD, BCE, ADE, ADF, AEF, and 7 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	2	0	0	2
3	7	5	5	7
4	5	5	5	5
5	0	2	2	0

Cross-certificate test The pair AB and ACD meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	75	25/56
2	63	3/8
3	28	1/6
4	2	1/84

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	19	2	0	0
3	19	11	0	5
4	5	11	0	19
5	0	2	0	19
6	0	0	0	7
7	0	0	0	1

One correlated failure Removing AB leaves $CDEFG$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 39 cases. For phase 2, A is pivotal in 39 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 51/128.

14.19 Q19. Weighted thresholds 9 and 10 over 8 nodes

Node weights in alphabetic order are $(4, 3, 3, 2, 2, 2, 1, 1)$, with total 18. The two weight thresholds are 9 and 10. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, ABD, ACD, ABE, ACE, ABF, ACF, and 26 more
Phase 2	ABC, $ABDE$, $ACDE$, $BCDE$, $ABDF$, $ACDF$, $BCDF$, and 26 more
Blocks 1	ABC, $ABDE$, $ACDE$, $BCDE$, $ABDF$, $ACDF$, $BCDF$, and 26 more
Blocks 2	ABC, ABD, ACD, ABE, ACE, ABF, ACF, and 26 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	7	1	1	7
4	20	22	22	20
5	6	10	10	6

Cross-certificate test The pair ABC and $ADEF$ meets in A. The smallest intersection has size 1. The fractions below

normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	242	2/9
2	507	169/363
3	290	290/1089
4	50	50/1089

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	28	0	0	0
3	49	6	0	1
4	27	16	0	27
5	1	6	0	49
6	0	0	0	28
7	0	0	0	8
8	0	0	0	1

One correlated failure Removing AB leaves $CDEFGH$. Availability is yes for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 64 cases. For phase 2, A is pivotal in 64 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 57/128.

14.20 Q20. Weighted thresholds 11 and 12 over 8 nodes

Node weights in alphabetic order are $(6, 5, 4, 3, 2, 1, 1, 1)$, with total 23. The two weight thresholds are 11 and 12. Failure of the heaviest nodes can remove more certificate capacity than the number of failures suggests.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, ACD, BCD, ACE, BCE, ADE, ACF, and 16 more
Phase 2	ABC, ABD, ACD, BCD, ABE, ACE, ABF, and 17 more
Blocks 1	ABC, ABD, ACD, ABE, $BCDE$, $BCDF$, $ACEF$, and 15 more
Blocks 2	ABC, ABD, ACD, BCD, ABE, ACE, ABF, and 17 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	1	0	0	0
3	8	9	4	9
4	9	9	10	9
5	5	5	7	5
6	0	1	1	1

Cross-certificate test The pair AB and $CDEFGH$ meets in $\varnothing$. The smallest intersection has size 0. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
0	20	5/138
1	189	63/184
2	223	223/552
3	94	47/276
4	23	1/24
5	3	1/184

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	27	1	0	0
3	42	5	0	9
4	26	9	0	35
5	4	5	0	47
6	0	0	0	28
7	0	0	0	8
8	0	0	0	1

One correlated failure Removing AB leaves $CDEFGH$. Availability is yes for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 68 cases. For phase 2, A is pivotal in 70 cases. If all survivor sets are equally likely, the probability of both certificates being possible is $1/2$.

14.21 Q21. Regional split 12/21

Regions have consecutive node groups of sizes $(2, 3)$. The first certificate requires region minima $(1, 2)$, and the second requires $(2, 1)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ACD, BCD, ACE, BCE, ADE, BDE
Phase 2	ABC, ABD, ABE
Blocks 1	AB, CD, CE, DE
Blocks 2	A, B, CDE

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	0	0	0	2
2	0	0	4	0
3	6	3	0	1

Cross-certificate test The pair ACD and ABE meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	6	1/3
2	12	2/3

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	10	0	0	0
3	1	6	3	0
4	0	2	0	3
5	0	0	0	1

One correlated failure Removing AB leaves CDE. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding C changes phase 1 from unavailable to available in 6 cases. For phase 2, A is pivotal in 7 cases. If all survivor sets are equally likely, the probability of both certificates being possible is $1/8$.

14.22 Q22. Regional split 12/21

Regions have consecutive node groups of sizes $(3, 3)$. The first certificate requires region minima $(1, 2)$, and the second requires

$(2, 1)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ADE, BDE, CDE, ADF, BDF, CDF, AEF, and 2 more
Phase 2	ABD, ACD, BCD, ABE, ACE, BCE, ABF, and 2 more
Blocks 1	DE, DF, EF, ABC
Blocks 2	AB, AC, BC, DEF

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	0	0	3	3
3	9	9	1	1

Cross-certificate test The pair ADE and BCF meets in $\varnothing$. The smallest intersection has size 0. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
0	9	1/9
1	36	4/9
2	36	4/9

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	15	0	0	0
3	2	9	9	0
4	0	3	3	9
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing ABC leaves DEF. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding D changes phase 1 from unavailable to available in 14 cases. For phase 2, A is pivotal in 14 cases. If all survivor sets are equally likely, the probability of both certificates being possible is $1/4$.

14.23 Q23. Regional split 22/21

Regions have consecutive node groups of sizes $(3, 3)$. The first certificate requires region minima $(2, 2)$, and the second requires $(2, 1)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ABDE$, $ACDE$, $BCDE$, $ABDF$, $ACDF$, $BCDF$, $ABEF$, and 2 more
Phase 2	ABD, ACD, BCD, ABE, ACE, BCE, ABF, and 2 more
Blocks 1	AB, AC, BC, DE, DF, EF
Blocks 2	AB, AC, BC, DEF

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	0	0	6	3
3	0	9	0	1
4	9	0	0	0

Cross-certificate test The pair $ABDE$ and ACF meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	18	2/9
2	45	5/9
3	18	2/9

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	15	0	0	0
3	11	0	9	0
4	3	0	3	9
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing ABC leaves DEF. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 8 cases. For phase 2, A is pivotal in 14 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/4.

14.24 Q24. Regional split 111/202

Regions have consecutive node groups of sizes $(2, 2, 3)$. The first certificate requires region minima $(1, 1, 1)$, and the second requires $(2, 0, 2)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ACE, BCE, ADE, BDE, ACF, BCF, ADF, and 5 more
Phase 2	$ABEF$, $ABEG$, $ABFG$
Blocks 1	AB, CD, EFG
Blocks 2	A, B, EF, EG, FG

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	0	0	0	2
2	0	0	2	3
3	12	0	1	0
4	0	3	0	0

Cross-certificate test The pair ACE and $ABFG$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	12	1/3
2	24	2/3

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	21	0	0	0
3	23	12	0	0
4	8	24	3	0
5	1	13	1	6
6	0	2	0	5
7	0	0	0	1

One correlated failure Removing AB leaves $CDEFG$. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase

1 from unavailable to available in 21 cases. For phase 2, A is pivotal in 16 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 3/32.

14.25 Q25. Regional split 121/212

Regions have consecutive node groups of sizes $(2, 3, 3)$. The first certificate requires region minima $(1, 2, 1)$, and the second requires $(2, 1, 2)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ACDF$, $BCDF$, $ACEF$, $BCEF$, $ADEF$, $BDEF$, $ACDG$, and 11 more
Phase 2	$ABCFG$, $ABDFG$, $ABEFG$, $ABCFH$, $ABDFH$, $ABEFH$, $ABCGH$, and 2 more
Blocks 1	AB, CD, CE, DE, FGH
Blocks 2	A, B, FG, FH, GH, CDE

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	0	0	0	2
2	0	0	4	3
3	0	0	1	1
4	18	0	0	0
5	0	9	0	0

Cross-certificate test The pair $ACDF$ and $ABEGH$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	18	1/9
2	72	4/9
3	72	4/9

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	28	0	0	0
3	56	0	0	0
4	52	18	0	0
5	14	33	9	0
6	1	15	3	9
7	0	2	0	6
8	0	0	0	1

One correlated failure Removing AB leaves $CDEFGH$. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding C changes phase 1 from unavailable to available in 42 cases. For phase 2, A is pivotal in 28 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/16.

14.26 Q26. Regional split 112/222

Regions have consecutive node groups of sizes $(3, 3, 3)$. The first certificate requires region minima $(1, 1, 2)$, and the second requires $(2, 2, 2)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ADGH$, $BDGH$, $CDGH$, $AEGH$, $BEGH$, $CEGH$, $AFGH$, and 20 more
Phase 2	$ABDEGH$, $ACDEGH$, $BCDEGH$, $ABDFGH$, $ACDFGH$, $BCDFGH$, $ABEFGH$, and 20 more
Blocks 1	GH, GI, HI, ABC, DEF
Blocks 2	AB, AC, BC, DE, DF, EF, GH, and 2 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	0	0	3	9
3	0	0	2	0
4	27	0	0	0
6	0	27	0	0

Cross-certificate test The pair $ADGH$ and $BCEFGI$ meets in G. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	54	2/27
2	243	1/3
3	324	4/9
4	108	4/27

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	9	0	0	0
2	36	0	0	0
3	84	0	0	0
4	99	27	0	0
5	63	63	0	0
6	21	36	0	27
7	3	6	0	27
8	0	0	0	9
9	0	0	0	1

One correlated failure Removing ABC leaves $DEFGHI$. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding G changes phase 1 from unavailable to available in 98 cases. For phase 2, A is pivotal in 32 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/8.

14.27 Q27. Regional split 1111/0202

Regions have consecutive node groups of sizes $(2, 2, 2, 2)$. The first certificate requires region minima $(1, 1, 1, 1)$, and the second requires $(0, 2, 0, 2)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ACEG$, $BCEG$, $ADEG$, $BDEG$, $ACFG$, $BCFG$, $ADFG$, and 9 more
Phase 2	$CDGH$
Blocks 1	AB, CD, EF, GH
Blocks 2	C, D, G, H

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	0	0	0	4
2	0	0	4	0
4	16	1	0	0

Cross-certificate test The pair $ACEG$ and $CDGH$ meets in CG. The smallest intersection has size 2. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
2	16	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	28	0	0	0
3	56	0	0	0
4	53	16	1	0
5	20	32	4	0
6	2	20	2	4
7	0	4	0	4
8	0	0	0	1

One correlated failure Removing AB leaves $CDEFGH$. Availability is no for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 27 cases. For phase 2, C is pivotal in 16 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 9/256.

14.28 Q28. Regional split 112/122

Regions have consecutive node groups of sizes $(1, 3, 4)$. The first certificate requires region minima $(1, 1, 2)$, and the second requires $(1, 2, 2)$. The named failure experiment removes the entire first region.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ABEF$, $ACEF$, $ADEF$, $ABEG$, $ACEG$, $ADEG$, $ABFG$, and 11 more
Phase 2	$ABCEF$, $ABDEF$, $ACDEF$, $ABCEG$, $ABDEG$, $ACDEG$, $ABCFG$, and 11 more
Blocks 1	A, BCD, EFG, EFH, EGH, FGH
Blocks 2	A, BC, BD, CD, EFG, EFH, EGH, and 1 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	0	0	1	1
2	0	0	0	3
3	0	0	5	4
4	18	0	0	0
5	0	18	0	0

Cross-certificate test The pair $ABEF$ and $ACDGH$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	18	1/18
2	108	1/3
3	162	1/2
4	36	1/9

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	28	0	0	0
3	56	0	0	0
4	52	18	0	0
5	26	12	0	18
6	7	3	0	18
7	1	0	0	7
8	0	0	0	1

One correlated failure Removing A leaves $BCDEFGH$. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 0 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 77 cases. For phase 2, A is pivotal in 44 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 11/64.

14.29 Q29. Row and column certificates on a 2 by 2 grid

A first certificate contains a whole row, and a second certificate contains a whole column. Nodes are named row by row. Every cross-pair intersects in one node, though distinct rows can be disjoint.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, CD
Phase 2	AC, BD
Blocks 1	AC, BC, AD, BD
Blocks 2	AB, BC, AD, CD

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	2	2	4	4

Cross-certificate test The pair AB and AC meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	4	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	4	0	0	0
2	2	2	2	0
3	0	0	0	4
4	0	0	0	1

One correlated failure Removing AB leaves CD. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 3 cases. For phase 2, A is pivotal in 3 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 5/16.

14.30 Q30. Row and column certificates on a 2 by 3 grid

A first certificate contains a whole row, and a second certificate contains a whole column. Nodes are named row by row. Every cross-pair intersects in one node, though distinct rows can be disjoint.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, DEF
Phase 2	AD, BE, CF
Blocks 1	AD, BD, CD, AE, BE, CE, AF, and 2 more
Blocks 2	ABC, BCD, ACE, CDE, ABF, BDF, AEF, and 1 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	0	3	9	0
3	2	0	0	8

Cross-certificate test The pair ABC and AD meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	6	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	12	0	3	0
3	6	2	12	0
4	0	0	9	6
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing ABC leaves DEF. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 7 cases. For phase 2, A is pivotal in 9 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 13/64.

14.31 Q31. Row and column certificates on a 3 by 2 grid

A first certificate contains a whole row, and a second certificate contains a whole column. Nodes are named row by row. Every cross-pair intersects in one node, though distinct rows can be disjoint.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, CD, EF
Phase 2	ACE, BDF
Blocks 1	ACE, BCE, ADE, BDE, ACF, BCF, ADF, and 1 more
Blocks 2	AB, BC, AD, CD, BE, DE, AF, and 2 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	3	0	0	9
3	0	2	8	0

Cross-certificate test The pair AB and ACE meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	6	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	12	3	0	0
3	6	12	2	0
4	0	9	0	6
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing AB leaves $CDEF$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 9 cases. For phase 2, A is pivotal in 7 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 13/64.

14.32 Q32. Row and column certificates on a 3 by 3 grid

A first certificate contains a whole row, and a second certificate contains a whole column. Nodes are named row by row. Every cross-pair intersects in one node, though distinct rows can be disjoint.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	ABC, DEF, GHI
Phase 2	ADG, BEH, CFI
Blocks 1	ADG, BDG, CDG, AEG, BEG, CEG, AFG, and 20 more
Blocks 2	ABC, BCD, ACE, CDE, ABF, BDF, AEF, and 20 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
3	3	3	27	27

Cross-certificate test The pair ABC and ADG meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	9	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	9	0	0	0
2	36	0	0	0
3	78	3	3	0
4	90	18	18	0
5	45	36	36	9
6	6	21	21	36
7	0	0	0	36
8	0	0	0	9
9	0	0	0	1

One correlated failure Removing ABC leaves $DEFGHI$. Availability is yes for

phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 2 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 49 cases. For phase 2, A is pivotal in 49 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 91/512.

14.33 Q33. Row and column certificates on a 2 by 4 grid

A first certificate contains a whole row, and a second certificate contains a whole column. Nodes are named row by row. Every cross-pair intersects in one node, though distinct rows can be disjoint.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	$ABCD$, $EFGH$
Phase 2	AE, BF, CG, DH
Blocks 1	AE, BE, CE, DE, AF, BF, CF, and 9 more
Blocks 2	$ABCD$, $BCDE$, $ACDF$, $CDEF$, $ABDG$, $BDEG$, $ADFG$, and 9 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	0	4	16	0
4	2	0	0	16

Cross-certificate test The pair $ABCD$ and AE meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	8	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	24	0	4	0
3	32	0	24	0
4	14	2	54	0
5	0	0	48	8
6	0	0	16	12
7	0	0	0	8
8	0	0	0	1

One correlated failure Removing $ABCD$ leaves $EFGH$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 1 and 3 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 15 cases. For phase 2, A is pivotal in 27 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 29/256.

14.34 Q34. Row and column certificates on a 4 by 2 grid

A first certificate contains a whole row, and a second certificate contains a whole column. Nodes are named row by row. Every cross-pair intersects in one node, though distinct rows can be disjoint.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, CD, EF, GH
Phase 2	$ACEG$, $BDFH$
Blocks 1	$ACEG$, $BCEG$, $ADEG$, $BDEG$, $ACFG$, $BCFG$, $ADFG$, and 9 more
Blocks 2	AB, BC, AD, CD, BE, DE, AF, and 9 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	4	0	0	16
4	0	2	16	0

Cross-certificate test The pair AB and $ACEG$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	8	1

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	24	4	0	0
3	32	24	0	0
4	14	54	2	0
5	0	48	0	8
6	0	16	0	12
7	0	0	0	8
8	0	0	0	1

One correlated failure Removing AB leaves $CDEFGH$. Availability is yes for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 27 cases. For phase 2, A is pivotal in 15 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 29/256.

14.35 Q35. 5-node ring with adjacent policy

First certificates are adjacent pairs on a ring. Second certificates are minimal vertex covers of that ring, so they hit every adjacent pair.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, BC, CD, AE, DE
Phase 2	ABD, ACD, ACE, BCE, BDE
Blocks 1	ABD, ACD, ACE, BCE, BDE
Blocks 2	AB, BC, CD, AE, DE

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	5	0	0	5
3	0	5	5	0

Cross-certificate test The pair AB and ACD meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	20	4/5
2	5	1/5

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	5	0	0	0
2	5	5	0	0
3	0	5	0	5
4	0	0	0	5
5	0	0	0	1

One correlated failure Removing ACE leaves BD. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 5 cases. For phase 2, A is pivotal in 5 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 11/32.

14.36 Q36. 6-node ring with adjacent policy

First certificates are adjacent pairs on a ring. Second certificates are minimal vertex covers of that ring, so they hit every adjacent pair.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, BC, CD, DE, AF, EF
Phase 2	ACE, BDF, $ABDE$, $ACDF$, $BCEF$
Blocks 1	ACE, BDF, $ABDE$, $ACDF$, $BCEF$
Blocks 2	AB, BC, CD, DE, AF, EF

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	6	0	0	6
3	0	2	2	0
4	0	3	3	0

Cross-certificate test The pair AB and ACE meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	24	4/5
2	6	1/5

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	6	0	0	0
2	9	6	0	0
3	0	18	2	0
4	0	6	0	9
5	0	0	0	6
6	0	0	0	1

One correlated failure Removing ACE leaves BDF. Availability is no for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 2 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 8 cases. For phase 2, A is pivotal in 8 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 1/4.

14.37 Q37. 7-node ring with adjacent policy

First certificates are adjacent pairs on a ring. Second certificates are minimal vertex covers of that ring, so they hit every adjacent pair.

Minimal certificate and cut census The following examples name actual node sets.

Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, BC, CD, DE, EF, AG, FG
Phase 2	$ABDF$, $ACDF$, $ACEF$, $ACEG$, $BCEG$, $BDEG$, $BDFG$
Blocks 1	$ABDF$, $ACDF$, $ACEF$, $ACEG$, $BCEG$, $BDEG$, $BDFG$
Blocks 2	AB, BC, CD, DE, EF, AG, FG

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	7	0	0	7
4	0	7	7	0

Cross-certificate test The pair AB and $ACDF$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	42	6/7
2	7	1/7

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	7	0	0	0
2	14	7	0	0
3	7	28	0	0
4	0	28	0	7
5	0	7	0	14
6	0	0	0	7
7	0	0	0	1

One correlated failure Removing $ACEG$ leaves BDF. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 13 cases. For phase 2, A is pivotal in 13 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 29/128.

14.38 Q38. 8-node ring with alternating policy

First certificates are adjacent pairs on a ring. A second certificate consists of one whole parity class in the linear labeling. The closing edge makes the distinction between even and odd rings.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, BC, CD, DE, EF, FG, AH, and 1 more
Phase 2	$ACEG$, $BDFH$
Blocks 1	$ACEG$, $BDFH$, $ABDEG$, $ABDFG$, $ACDFG$, $ACDFH$, $ACEFH$, and 3 more
Blocks 2	AB, BC, AD, CD, BE, DE, AF, and 9 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	8	0	0	16
4	0	2	2	0
5	0	0	8	0

Cross-certificate test The pair AB and $ACEG$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	16	1

Survivor profile The middle columns

record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	8	0	0	0
2	20	8	0	0
3	16	40	0	0
4	0	68	2	0
5	0	48	0	8
6	0	16	0	12
7	0	0	0	8
8	0	0	0	1

One correlated failure Removing $ACEG$ leaves $BDFH$. Availability is no for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 3 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 21 cases. For phase 2, A is pivotal in 15 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 29/256.

14.39 Q39. 9-node ring with alternating policy

First certificates are adjacent pairs on a ring. A second certificate consists of one whole parity class in the linear labeling. The closing edge makes the distinction between even and odd rings.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, BC, CD, DE, EF, FG, GH, and 2 more
Phase 2	$BDFH$, $ACEGI$
Blocks 1	$ABDFH$, $ACDFH$, $ACEFH$, $ACEGH$, $ACEGI$, $BCEGI$, $BDEGI$, and 5 more
Blocks 2	AB, BC, AD, CD, BE, DE, AF, and 13 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
2	9	0	0	20
4	0	1	0	0
5	0	1	9	0
6	0	0	3	0

Cross-certificate test The pair AI and $BDFH$ meets in $\varnothing$. The smallest intersection has size 0. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
0	1	1/18
1	16	8/9
2	1	1/18

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	9	0	0	0
2	27	9	0	0
3	30	54	0	0
4	8	117	1	0
5	0	120	0	6
6	0	70	0	14
7	0	20	0	16
8	0	0	0	9
9	0	0	0	1

One correlated failure Removing $ACEGI$

leaves $BDFH$. Availability is no for phase 1 and yes for phase 2. Over all failure sets the sharp uniform tolerances are 4 and 1 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 34 cases. For phase 2, B is pivotal in 31 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 23/256.

14.40 Q40. 9-node ring with anchored policy

First certificates are adjacent pairs on a ring. Second certificates hit all edges and also include A. The added anchor has a separate effect on availability.

Minimal certificate and cut census The following examples name actual node sets. Minimal certificate counts and minimal blocking-set counts are grouped by size.

Object	Examples
Phase 1	AB, BC, CD, DE, EF, FG, GH, and 2 more
Phase 2	$ABDFH$, $ACDFH$, $ACEFH$, $ACEGH$, $ACEGI$, $ABDEGH$, $ABDEGI$, and 2 more
Blocks 1	$ABDFH$, $ACDFH$, $ACEFH$, $ACEGH$, $ACEGI$, $BCEGI$, $BDEGI$, and 5 more
Blocks 2	A, BC, CD, DE, EF, FG, GH, and 1 more

Size	Phase 1	Phase 2	Cuts 1	Cuts 2
1	0	0	0	1
2	9	0	0	7
5	0	5	9	0
6	0	4	3	0

Cross-certificate test The pair AB and $ACDFH$ meets in A. The smallest intersection has size 1. The fractions below normalize by the number of ordered cross-pairs.

Overlap	Pairs	Fraction
1	64	64/81
2	17	17/81

Survivor profile The middle columns record availability of exactly one family. The last column records simultaneous availability. Entries in a row sum to the binomial count of survivor sets.

Live	Neither	Only 1	Only 2	Both
0	1	0	0	0
1	9	0	0	0
2	27	9	0	0
3	30	54	0	0
4	9	117	0	0
5	0	121	0	5
6	0	64	0	20
7	0	15	0	21
8	0	1	0	8
9	0	0	0	1

One correlated failure Removing $ACEGI$ leaves $BDFH$. Availability is no for phase 1 and no for phase 2. Over all failure sets the sharp uniform tolerances are 4 and 0 failures, respectively.

Pivotal observations Among subsets of the other nodes, adding A changes phase 1 from unavailable to available in 34 cases. For phase 2, A is pivotal in 55 cases. If all survivor sets are equally likely, the probability of both certificates being possible is 55/512.

15 Atlas of timed boundary constraints

Every row $U, V, [\ell, u]$ means $\ell \le t_V - t_U \le u$, with integer times in arbitrary units. The weighted graph has edges $U \to V$ of weight u and $V \to U$ of weight $-\ell$. Entries

in the distance matrix are shortest-path upper bounds. Theorem 7.6 converts them into tight boundary intervals. The cases use three distinct instrumentation profiles for each network geometry. Each supplies both realizing schedules and a negative-cycle certificate for a stricter boundary.

15.1 T01. Serial relays, uniform instrumentation

Each local calibration concerns successive relays. The additional intervals compare an earlier relay with a later one across an intermediate stage.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 6]$	primary
B	C	$[6, 9]$	primary
C	D	$[2, 5]$	primary
D	E	$[5, 7]$	primary
A	C	$[10, 13]$	calibration
B	D	$[10, 12]$	calibration
C	E	$[9, 12]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 6 & 13 & 18 & 25 \\ -4 & 0 & 9 & 12 & 19 \\ -10 & -6 & 0 & 5 & 12 \\ -14 & -10 & -2 & 0 & 7 \\ -19 & -15 & -9 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 19 to 25. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	4	10	14	19
Upper	0	6	13	18	25

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to E$	25	A, E
$E \to C \to A$	-19	E, A

Changed observations Pinning $t_C = 11$ gives endpoint interval $[20, 23]$. Removing the comparison on A–C with interval $[10, 13]$ gives $[19, 25]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 18$. This contributes an edge from A to the endpoint. Follow it by $E \to C \to A$. The resulting cycle has total weight $(18) + (-19) = -1$, so no completion satisfies the deadline.

15.2 T02. Serial relays, tight cross-window

Each local calibration concerns successive relays. The additional intervals compare an earlier relay with a later one across an intermediate stage.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 9]$	primary
B	C	$[1, 5]$	primary
C	D	$[2, 8]$	primary
D	E	$[5, 9]$	primary
A	C	$[9, 9]$	calibration
B	D	$[8, 8]$	calibration
C	E	$[11, 13]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 9 & 16 & 22 \\ -4 & 0 & 5 & 8 & 17 \\ -9 & -1 & 0 & 7 & 13 \\ -12 & -8 & -3 & 0 & 9 \\ -20 & -13 & -11 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 20 to 22. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	7	9	15	20
Upper	0	8	9	16	22

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to E$	22	A, E
$E \to C \to A$	-20	E, A

Changed observations Pinning $t_C = 9$ gives endpoint interval $[20, 22]$. Removing the comparison on A–C with interval $[9, 9]$ gives $[16, 26]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 19$. This contributes an edge from A to the endpoint. Follow it by $E \to C \to A$. The resulting cycle has total weight $(19) + (-20) = -1$, so no completion satisfies the deadline.

15.3 T03. Serial relays, asymmetric delivery

Each local calibration concerns successive relays. The additional intervals compare an earlier relay with a later one across an intermediate stage.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[6, 10]$	primary
B	C	$[0, 5]$	primary
C	D	$[6, 10]$	primary
D	E	$[0, 3]$	primary
A	C	$[9, 13]$	calibration
B	D	$[10, 12]$	calibration
C	E	$[7, 9]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 10 & 13 & 22 & 22 \\ -6 & 0 & 5 & 12 & 14 \\ -9 & -1 & 0 & 9 & 9 \\ -16 & -10 & -6 & 0 & 3 \\ -16 & -10 & -7 & 0 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 16 to 22. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	6	9	16	16
Upper	0	10	13	22	22

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to E$	22	A, E
$E \to C \to A$	-16	E, A

Changed observations Pinning $t_C = 11$ gives endpoint interval $[18, 20]$. Removing the comparison on C–E with interval $[7, 9]$ gives $[16, 25]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 15$. This contributes an edge from A to the endpoint. Follow it by $E \to C \to A$. The resulting cycle has total weight $(15) + (-16) = -1$, so no completion satisfies the deadline.

15.4 T04. Wide replication fan, uniform instrumentation

Four processing branches start after a shared dispatch. A joint receipt depends on all branches. The optional calibration compares siblings across links that need not impose a causal order.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[5, 8]$	primary
A	C	$[8, 10]$	primary
A	D	$[12, 15]$	primary
A	E	$[20, 22]$	primary
B	F	$[18, 21]$	primary
C	F	$[15, 17]$	primary
D	F	$[9, 12]$	primary
E	F	$[3, 5]$	primary
B	C	$[2, 5]$	calibration
C	D	$[4, 6]$	calibration
D	E	$[5, 8]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 10 & 15 & 22 & 27 \\ -5 & 0 & 5 & 10 & 17 & 21 \\ -8 & -2 & 0 & 6 & 14 & 17 \\ -12 & -6 & -4 & 0 & 8 & 12 \\ -20 & -13 & -10 & -5 & 0 & 5 \\ -23 & -18 & -15 & -9 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 23 to 27. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	5	8	14	20	23
Upper	0	8	10	15	22	27

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to F$	27	A, F
$F \to B \to A$	-23	F, A

Changed observations Pinning $t_D = 13$ gives endpoint interval $[23, 25]$. Removing the comparison on B–C with interval $[2, 5]$ gives $[23, 27]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \leq 22$. This contributes an edge from A to the endpoint. Follow it by $F \to B \to A$. The resulting cycle has total weight $(22) + (-23) = -1$, so no completion satisfies the deadline.

15.5 T05. Wide replication fan, tight cross-window

Four processing branches start after a shared dispatch. A joint receipt depends on all branches. The optional calibration compares siblings across links that need not impose a causal order.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 9]$	primary
A	C	$[9, 14]$	primary
A	D	$[14, 19]$	primary
A	E	$[18, 23]$	primary
B	F	$[15, 20]$	primary
C	F	$[12, 17]$	primary
D	F	$[5, 10]$	primary
E	F	$[3, 8]$	primary
B	C	$[4, 5]$	calibration
C	D	$[5, 6]$	calibration
D	E	$[3, 3]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 9 & 14 & 19 & 22 & 29 \\ -4 & 0 & 5 & 11 & 14 & 20 \\ -9 & -4 & 0 & 6 & 9 & 16 \\ -15 & -9 & -5 & 0 & 3 & 10 \\ -18 & -12 & -8 & -3 & 0 & 7 \\ -21 & -16 & -12 & -6 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 21 to 29. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	5	9	15	18	21
Upper	0	9	14	19	22	29

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to B \to F$	29	A, F
$F \to C \to A$	-21	F, A

Changed observations Pinning $t_D = 17$ gives endpoint interval $[23, 27]$. Removing the comparison on B–C with interval $[4, 5]$ gives $[21, 29]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 20$. This contributes an edge from A to the endpoint. Follow it by $F \to C \to A$. The resulting cycle has total weight $(20) + (-21) = -1$, so no completion satisfies the deadline.

15.6 T06. Wide replication fan, asymmetric delivery

Four processing branches start after a shared dispatch. A joint receipt depends on all branches. The optional calibration compares siblings across links that need not impose a causal order.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 4]$	primary
A	C	$[4, 12]$	primary
A	D	$[15, 15]$	primary
A	E	$[16, 22]$	primary
B	F	$[21, 23]$	primary
C	F	$[13, 21]$	primary
D	F	$[10, 10]$	primary
E	F	$[3, 9]$	primary
B	C	$[5, 5]$	calibration
C	D	$[5, 9]$	calibration
D	E	$[2, 6]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 4 & 9 & 15 & 21 & 25 \\ -2 & 0 & 5 & 13 & 19 & 23 \\ -7 & -5 & 0 & 8 & 14 & 18 \\ -15 & -11 & -6 & 0 & 6 & 10 \\ -17 & -13 & -8 & -2 & 0 & 8 \\ -25 & -21 & -16 & -10 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 25 to 25. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	4	9	15	21	25
Upper	0	4	9	15	21	25

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to D \to F$	25	A, F
$F \to D \to A$	-25	F, A

Changed observations Pinning $t_D = 15$ gives endpoint interval $[25, 25]$. Removing the comparison on B–C with interval $[5, 5]$ gives $[25, 25]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 24$. This contributes an edge from A to the endpoint. Follow it by $F \to D \to A$. The resulting cycle has total weight $(24) + (-25) = -1$, so no completion satisfies the deadline.

15.7 T07. Narrow join, uniform instrumentation

Two prepare paths meet before a final store. The cross-branch interval comes from an observation of both prepare paths.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[6, 8]$	primary
A	C	$[10, 12]$	primary
B	D	$[8, 11]$	primary
C	D	$[5, 8]$	primary
D	E	$[2, 4]$	primary
B	C	$[3, 5]$	calibration
A	D	$[16, 18]$	calibration
C	E	$[7, 11]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 12 & 18 & 22 \\ -6 & 0 & 5 & 11 & 15 \\ -10 & -3 & 0 & 8 & 11 \\ -16 & -8 & -5 & 0 & 4 \\ -18 & -10 & -7 & -2 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 18 to 22. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	8	11	16	18
Upper	0	8	12	18	22

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to D \to E$	22	A, E
$E \to D \to A$	-18	E, A

Changed observations Pinning $t_C = 11$ gives endpoint interval $[18, 22]$. Removing the comparison on A–D with interval $[16, 18]$ gives $[17, 23]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 17$. This contributes an edge from A to the endpoint. Follow it by $E \to D \to A$. The resulting cycle has total weight $(17) + (-18) = -1$, so no completion satisfies the deadline.

15.8 T08. Narrow join, tight cross-window

Two prepare paths meet before a final store. The cross-branch interval comes from an observation of both prepare paths.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[0, 6]$	primary
A	C	$[6, 10]$	primary
B	D	$[9, 15]$	primary
C	D	$[5, 9]$	primary
D	E	$[1, 7]$	primary
B	C	$[5, 5]$	calibration
A	D	$[14, 15]$	calibration
C	E	$[11, 12]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 5 & 10 & 15 & 22 \\ -1 & 0 & 5 & 14 & 17 \\ -6 & -5 & 0 & 9 & 12 \\ -14 & -10 & -5 & 0 & 7 \\ -17 & -16 & -11 & -2 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 17 to 22. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	1	6	15	17
Upper	0	5	10	15	22

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to E$	22	A, E
$E \to C \to A$	-17	E, A

Changed observations Pinning $t_C = 8$ gives endpoint interval $[19, 20]$. Removing the comparison on C–E with interval $[11, 12]$ gives $[15, 22]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 16$. This contributes an edge from A to the endpoint. Follow it by $E \to C \to A$. The resulting cycle has total weight $(16) + (-17) = -1$, so no completion satisfies the deadline.

15.9 T09. Narrow join, asymmetric delivery

Two prepare paths meet before a final store. The cross-branch interval comes from an observation of both prepare paths.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 8]$	primary
A	C	$[6, 10]$	primary
B	D	$[9, 12]$	primary
C	D	$[0, 4]$	primary
D	E	$[4, 9]$	primary
B	C	$[6, 8]$	calibration
A	D	$[11, 15]$	calibration
C	E	$[6, 8]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 4 & 10 & 14 & 18 \\ -3 & 0 & 7 & 11 & 15 \\ -9 & -6 & 0 & 4 & 8 \\ -12 & -9 & -2 & 0 & 6 \\ -16 & -13 & -6 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 16 to 18. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	3	10	12	16
Upper	0	4	10	14	18

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to E$	18	A, E
$E \to D \to B \to A$	-16	E, A

Changed observations Pinning $t_C = 9$ gives endpoint interval $[16, 17]$. Removing the comparison on C–E with interval $[6, 8]$ gives $[16, 23]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 15$. This contributes an edge from A to the endpoint. Follow it by $E \to D \to B \to A$. The resulting cycle

has total weight $(15) + (-16) = -1$, so no completion satisfies the deadline.

15.10 T10. Two-tier collection, uniform instrumentation

Two collection nodes observe overlapping requests. The final certificate depends on both collection nodes.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 4]$	primary
A	C	$[7, 9]$	primary
A	D	$[13, 17]$	primary
B	E	$[15, 17]$	primary
C	E	$[10, 12]$	primary
C	F	$[16, 18]$	primary
D	F	$[8, 12]$	primary
E	G	$[8, 10]$	primary
F	G	$[2, 4]$	primary
B	D	$[11, 14]$	calibration
E	F	$[4, 7]$	calibration
A	G	$[27, 29]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 4 & 9 & 17 & 21 & 27 & 29 \\ -2 & 0 & 7 & 14 & 17 & 24 & 27 \\ -7 & -3 & 0 & 10 & 12 & 18 & 22 \\ -13 & -11 & -4 & 0 & 6 & 12 & 16 \\ -17 & -15 & -10 & -1 & 0 & 7 & 10 \\ -23 & -19 & -16 & -8 & -4 & 0 & 4 \\ -27 & -23 & -18 & -10 & -8 & -2 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 27 to 29. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	4	9	17	19	25	27
Upper	0	4	9	17	21	27	29

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	29	A, G
$G \to A$	-27	G, A

Changed observations Pinning $t_D = 15$ gives endpoint interval $[27, 29]$. Removing the comparison on A–G with interval $[27, 29]$ gives $[25, 31]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 26$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(26) + (-27) = -1$, so no completion satisfies the deadline.

15.11 T11. Two-tier collection, tight cross-window

Two collection nodes observe overlapping requests. The final certificate depends on both collection nodes.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[1, 6]$	primary
A	C	$[8, 13]$	primary
A	D	$[10, 15]$	primary
B	E	$[12, 17]$	primary
C	E	$[5, 10]$	primary
C	F	$[13, 18]$	primary
D	F	$[9, 14]$	primary
E	G	$[9, 14]$	primary
F	G	$[1, 6]$	primary
B	D	$[8, 10]$	calibration
E	F	$[7, 7]$	calibration
A	G	$[29, 29]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-

diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 6 & 13 & 15 & 20 & 27 & 29 \\ -1 & 0 & 11 & 10 & 17 & 24 & 28 \\ -8 & -2 & 0 & 7 & 10 & 17 & 21 \\ -10 & -8 & 1 & 0 & 7 & 14 & 19 \\ -16 & -12 & -6 & -2 & 0 & 7 & 13 \\ -23 & -19 & -13 & -9 & -7 & 0 & 6 \\ -29 & -23 & -16 & -14 & -9 & -2 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 29 to 29. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	6	13	15	20	27	29
Upper	0	6	13	15	20	27	29

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	29	A, G
$G \to A$	-29	G, A

Changed observations Pinning $t_D = 12$ gives endpoint interval $[29, 29]$. Removing the comparison on A–G with interval $[29, 29]$ gives $[23, 35]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 28$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(28) + (-29) = -1$, so no completion satisfies the deadline.

15.12 T12. Two-tier collection, asymmetric delivery

Two collection nodes observe overlapping requests. The final certificate depends on both collection nodes.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 5]$	primary
A	C	$[8, 15]$	primary
A	D	$[16, 17]$	primary
B	E	$[14, 21]$	primary
C	E	$[9, 10]$	primary
C	F	$[9, 16]$	primary
D	F	$[9, 10]$	primary
E	G	$[5, 12]$	primary
F	G	$[4, 5]$	primary
B	D	$[9, 11]$	calibration
E	F	$[1, 5]$	calibration
A	G	$[30, 32]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 5 & 15 & 16 & 25 & 26 & 31 \\ -5 & 0 & 10 & 11 & 20 & 21 & 26 \\ -10 & -5 & 0 & 6 & 10 & 15 & 20 \\ -16 & -11 & -1 & 0 & 9 & 10 & 15 \\ -20 & -15 & -9 & -4 & 0 & 5 & 10 \\ -25 & -20 & -10 & -9 & -1 & 0 & 5 \\ -30 & -25 & -15 & -14 & -5 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 30 to 31. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	5	15	16	25	26	30
Upper	0	5	15	16	25	26	31

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to B \to D \to F \to G$	31	A, G
$G \to A$	-30	G, A

Changed observations Pinning $t_D = 16$ gives endpoint interval $[30, 31]$. Removing the comparison on B–D with interval $[9, 11]$

gives $[30, 32]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 29$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(29) + (-30) = -1$, so no completion satisfies the deadline.

15.13 T13. Paired pipeline, uniform instrumentation

Two three-step workers exchange calibration observations at corresponding stages. The additional observations cut diagonally across the worker-local chains.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 5]$	primary
B	C	$[5, 8]$	primary
D	E	$[3, 6]$	primary
E	F	$[6, 8]$	primary
A	D	$[12, 14]$	primary
B	E	$[13, 16]$	primary
C	F	$[13, 16]$	primary
A	E	$[16, 19]$	calibration
B	F	$[20, 23]$	calibration
C	F	$[14, 16]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 5 & 13 & 14 & 19 & 27 \\ -3 & 0 & 8 & 11 & 16 & 23 \\ -8 & -5 & 0 & 6 & 10 & 16 \\ -12 & -7 & 0 & 0 & 6 & 14 \\ -16 & -13 & -6 & -3 & 0 & 8 \\ -23 & -20 & -14 & -9 & -6 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 23 to 27. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	3	9	14	17	23
Upper	0	5	13	14	19	27

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to E \to F$	27	A, F
$F \to B \to A$	-23	F, A

Changed observations Pinning $t_D = 13$ gives endpoint interval $[23, 27]$. Removing the comparison on A–E with interval $[16, 19]$ gives $[23, 28]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 22$. This contributes an edge from A to the endpoint. Follow it by $F \to B \to A$. The resulting cycle has total weight $(22) + (-23) = -1$, so no completion satisfies the deadline.

15.14 T14. Paired pipeline, tight cross-window

Two three-step workers exchange calibration observations at corresponding stages. The additional observations cut diagonally across the worker-local chains.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 8]$	primary
B	C	$[5, 9]$	primary
D	E	$[3, 9]$	primary
E	F	$[1, 5]$	primary
A	D	$[13, 19]$	primary
B	E	$[15, 19]$	primary
C	F	$[10, 16]$	primary
A	E	$[22, 22]$	calibration
B	F	$[20, 21]$	calibration
C	F	$[12, 13]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 15 & 19 & 22 & 27 \\ -3 & 0 & 9 & 16 & 19 & 21 \\ -10 & -7 & 0 & 9 & 12 & 13 \\ -13 & -6 & 2 & 0 & 9 & 14 \\ -22 & -15 & -7 & -3 & 0 & 5 \\ -23 & -20 & -12 & -4 & -1 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 23 to 27. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	3	11	19	22	23
Upper	0	7	15	19	22	27

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to E \to F$	27	A, F
$F \to E \to A$	-23	F, A

Changed observations Pinning $t_D = 16$ gives endpoint interval $[23, 27]$. Removing the comparison on A–E with interval $[22, 22]$ gives $[22, 29]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 22$. This contributes an edge from A to the endpoint. Follow it by $F \to E \to A$. The resulting cycle has total weight $(22) + (-23) = -1$, so no completion satisfies the deadline.

15.15 T15. Paired pipeline, asymmetric delivery

Two three-step workers exchange calibration observations at corresponding stages. The additional observations cut diagonally across the worker-local chains.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[5, 9]$	primary
B	C	$[-1, 4]$	primary
D	E	$[7, 11]$	primary
E	F	$[1, 4]$	primary
A	D	$[13, 17]$	primary
B	E	$[11, 16]$	primary
C	F	$[16, 20]$	primary
A	E	$[19, 23]$	calibration
B	F	$[19, 19]$	calibration
C	F	$[14, 18]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 11 & 16 & 23 & 27 \\ -5 & 0 & 3 & 9 & 16 & 19 \\ -6 & -1 & 0 & 8 & 15 & 18 \\ -13 & -5 & -2 & 0 & 10 & 14 \\ -20 & -15 & -12 & -7 & 0 & 4 \\ -24 & -19 & -16 & -10 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 24 to 27. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	5	8	14	21	24
Upper	0	8	11	16	23	27

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to E \to F$	27	A, F
$F \to B \to A$	-24	F, A

Changed observations Pinning $t_D = 14$ gives endpoint interval $[24, 27]$. Removing the comparison on B–F with interval

$[19, 19]$ gives $[21, 27]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 23$. This contributes an edge from A to the endpoint. Follow it by $F \to B \to A$. The resulting cycle has total weight $(23) + (-24) = -1$, so no completion satisfies the deadline.

15.16 T16. Asymmetric pipeline, uniform instrumentation

A delayed read depends on earlier serialized writes. Long baseline intervals constrain more than one local delay at once.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 7]$	primary
B	C	$[6, 8]$	primary
C	D	$[2, 5]$	primary
D	E	$[5, 7]$	primary
E	F	$[2, 5]$	primary
F	G	$[4, 6]$	primary
A	E	$[21, 23]$	calibration
B	F	$[18, 22]$	calibration
C	G	$[17, 19]$	calibration
A	G	$[29, 31]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 14 & 18 & 23 & 27 & 31 \\ -4 & 0 & 8 & 13 & 19 & 22 & 27 \\ -10 & -6 & 0 & 5 & 12 & 15 & 19 \\ -14 & -8 & -2 & 0 & 7 & 12 & 17 \\ -21 & -14 & -7 & -5 & 0 & 5 & 10 \\ -23 & -18 & -11 & -7 & -2 & 0 & 6 \\ -29 & -23 & -17 & -12 & -6 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 29 to 31. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	6	12	17	23	25	29
Upper	0	7	14	18	23	27	31

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	31	A, G
$G \to A$	-29	G, A

Changed observations Pinning $t_D = 16$ gives endpoint interval $[29, 31]$. Removing the comparison on A–G with interval $[29, 31]$ gives $[27, 34]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 28$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(28) + (-29) = -1$, so no completion satisfies the deadline.

15.17 T17. Asymmetric pipeline, tight cross-window

A delayed read depends on earlier serialized writes. Long baseline intervals constrain more than one local delay at once.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 8]$	primary
B	C	$[1, 6]$	primary
C	D	$[2, 7]$	primary
D	E	$[5, 10]$	primary
E	F	$[1, 6]$	primary
F	G	$[4, 9]$	primary
A	E	$[20, 21]$	calibration
B	F	$[19, 20]$	calibration
C	G	$[22, 22]$	calibration
A	G	$[30, 31]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 9 & 16 & 21 & 27 & 31 \\ -3 & 0 & 6 & 13 & 18 & 20 & 28 \\ -8 & -1 & 0 & 7 & 13 & 18 & 22 \\ -10 & -3 & -2 & 0 & 10 & 16 & 20 \\ -20 & -13 & -11 & -5 & 0 & 6 & 11 \\ -22 & -19 & -13 & -6 & -1 & 0 & 9 \\ -30 & -23 & -22 & -15 & -9 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 30 to 31. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	7	8	15	21	26	30
Upper	0	8	9	16	21	27	31

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	31	A, G
$G \to A$	-30	G, A

Changed observations Pinning $t_D = 13$ gives endpoint interval $[30, 31]$. Removing the comparison on A–G with interval $[30, 31]$ gives $[26, 36]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 29$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(29) + (-30) = -1$, so no completion satisfies the deadline.

15.18 T18. Asymmetric pipeline, asymmetric delivery

A delayed read depends on earlier serialized writes. Long baseline intervals constrain more than one local delay at once.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[6, 8]$	primary
B	C	$[0, 8]$	primary
C	D	$[6, 6]$	primary
D	E	$[0, 6]$	primary
E	F	$[4, 6]$	primary
F	G	$[3, 11]$	primary
A	E	$[18, 22]$	calibration
B	F	$[16, 18]$	calibration
C	G	$[21, 23]$	calibration
A	G	$[30, 34]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 13 & 19 & 22 & 26 & 34 \\ -6 & 0 & 7 & 13 & 14 & 18 & 28 \\ -7 & 0 & 0 & 6 & 12 & 18 & 23 \\ -13 & -6 & -6 & 0 & 6 & 12 & 17 \\ -18 & -10 & -6 & 0 & 0 & 6 & 16 \\ -22 & -16 & -10 & -4 & -4 & 0 & 11 \\ -30 & -22 & -21 & -15 & -9 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 30 to 34. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	8	9	15	21	26	30
Upper	0	8	13	19	22	26	34

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	34	A, G
$G \to A$	-30	G, A

Changed observations Pinning $t_D = 16$ gives endpoint interval $[31, 33]$. Removing the comparison on A–G with interval

$[30, 34]$ gives $[27, 37]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 29$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(29) + (-30) = -1$, so no completion satisfies the deadline.

15.19 T19. Staged recovery, uniform instrumentation

A boot observation starts two recovery stages, each with a durable checkpoint. One service-ready observation joins them.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[5, 7]$	primary
B	C	$[2, 4]$	primary
A	D	$[12, 15]$	primary
D	E	$[6, 9]$	primary
C	F	$[15, 17]$	primary
E	F	$[3, 5]$	primary
B	D	$[7, 10]$	calibration
C	E	$[10, 13]$	calibration
A	F	$[24, 26]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 11 & 15 & 23 & 26 \\ -5 & 0 & 4 & 10 & 17 & 21 \\ -7 & -2 & 0 & 7 & 13 & 17 \\ -12 & -7 & -3 & 0 & 9 & 14 \\ -19 & -13 & -10 & -6 & 0 & 5 \\ -24 & -17 & -15 & -9 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 24 to 26. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	7	9	15	21	24
Upper	0	7	11	15	23	26

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to F$	26	A, F
$F \to A$	-24	F, A

Changed observations Pinning $t_D = 13$ gives endpoint interval $[24, 26]$. Removing the comparison on A–F with interval $[24, 26]$ gives $[22, 28]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 23$. This contributes an edge from A to the endpoint. Follow it by $F \to A$. The resulting cycle has total weight $(23) + (-24) = -1$, so no completion satisfies the deadline.

15.20 T20. Staged recovery, tight cross-window

A boot observation starts two recovery stages, each with a durable checkpoint. One service-ready observation joins them.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 10]$	primary
B	C	$[2, 6]$	primary
A	D	$[14, 20]$	primary
D	E	$[1, 5]$	primary
C	F	$[11, 17]$	primary
E	F	$[3, 7]$	primary
B	D	$[9, 11]$	calibration
C	E	$[9, 9]$	calibration
A	F	$[25, 25]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 10 & 13 & 20 & 22 & 25 \\ -4 & 0 & 6 & 11 & 15 & 21 \\ -9 & -2 & 0 & 8 & 9 & 16 \\ -14 & -9 & -4 & 0 & 5 & 11 \\ -18 & -11 & -9 & -1 & 0 & 7 \\ -25 & -15 & -12 & -5 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 25 to 25. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	10	13	20	22	25
Upper	0	10	13	20	22	25

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to F$	25	A, F
$F \to A$	-25	F, A

Changed observations Pinning $t_D = 17$ gives endpoint interval $[25, 25]$. Removing the comparison on A–F with interval $[25, 25]$ gives $[18, 32]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 24$. This contributes an edge from A to the endpoint. Follow it by $F \to A$. The resulting cycle has total weight $(24) + (-25) = -1$, so no completion satisfies the deadline.

15.21 T21. Staged recovery, asymmetric delivery

A boot observation starts two recovery stages, each with a durable checkpoint. One service-ready observation joins them.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 7]$	primary
B	C	$[1, 5]$	primary
A	D	$[15, 18]$	primary
D	E	$[1, 5]$	primary
C	F	$[16, 21]$	primary
E	F	$[2, 6]$	primary
B	D	$[10, 12]$	calibration
C	E	$[9, 13]$	calibration
A	F	$[25, 27]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 11 & 18 & 23 & 27 \\ -3 & 0 & 5 & 12 & 17 & 23 \\ -6 & -1 & 0 & 11 & 13 & 19 \\ -15 & -10 & -5 & 0 & 5 & 11 \\ -19 & -12 & -10 & -1 & 0 & 6 \\ -25 & -18 & -16 & -7 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 25 to 27. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	7	9	18	22	25
Upper	0	7	11	18	23	27

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to F$	27	A, F
$F \to A$	-25	F, A

Changed observations Pinning $t_D = 16$ gives endpoint interval $[25, 27]$. Removing the comparison on A–F with interval $[25, 27]$ gives $[20, 29]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 24$. This contributes an edge from A to the endpoint. Follow it by $F \to A$. The resulting cycle has total weight $(24) + (-25) = -1$, so no completion satisfies the deadline.

15.22 T22. Snapshot boundary, uniform instrumentation

The paired marker observations bracket two channel records, followed by a shared snapshot boundary and an export event.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[6, 8]$	primary
A	C	$[10, 12]$	primary
B	D	$[8, 12]$	primary
C	E	$[8, 10]$	primary
D	F	$[7, 9]$	primary
E	F	$[4, 6]$	primary
F	G	$[5, 9]$	primary
B	C	$[2, 5]$	calibration
D	E	$[2, 5]$	calibration
A	G	$[31, 33]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 12 & 20 & 22 & 28 & 33 \\ -6 & 0 & 5 & 12 & 15 & 21 & 27 \\ -10 & -2 & 0 & 8 & 10 & 16 & 23 \\ -14 & -8 & -3 & 0 & 5 & 9 & 18 \\ -18 & -10 & -8 & -2 & 0 & 6 & 15 \\ -22 & -15 & -12 & -7 & -4 & 0 & 9 \\ -31 & -23 & -19 & -12 & -9 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 31 to 33. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	8	12	19	22	26	31
Upper	0	8	12	20	22	28	33

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	33	A, G
$G \to A$	-31	G, A

Changed observations Pinning $t_D = 17$ gives endpoint interval $[31, 33]$. Removing the comparison on A–G with interval $[31, 33]$ gives $[27, 37]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 30$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(30) + (-31) = -1$, so no completion satisfies the deadline.

15.23 T23. Snapshot boundary, tight cross-window

The paired marker observations bracket two channel records, followed by a shared snapshot boundary and an export event.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[0, 5]$	primary
A	C	$[6, 11]$	primary
B	D	$[9, 14]$	primary
C	E	$[9, 14]$	primary
D	F	$[7, 12]$	primary
E	F	$[4, 9]$	primary
F	G	$[0, 5]$	primary
B	C	$[5, 5]$	calibration
D	E	$[4, 5]$	calibration
A	G	$[27, 28]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 5 & 10 & 19 & 24 & 28 & 28 \\ -1 & 0 & 5 & 14 & 19 & 26 & 27 \\ -6 & -5 & 0 & 9 & 14 & 21 & 22 \\ -10 & -9 & -4 & 0 & 5 & 12 & 17 \\ -15 & -14 & -9 & -4 & 0 & 8 & 13 \\ -22 & -18 & -13 & -8 & -4 & 0 & 5 \\ -27 & -22 & -17 & -8 & -4 & 0 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 27 to 28. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	5	10	19	23	27	27
Upper	0	5	10	19	24	28	28

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	28	A, G
$G \to A$	-27	G, A

Changed observations Pinning $t_D = 14$ gives endpoint interval $[27, 28]$. Removing the comparison on A–G with interval $[27, 28]$ gives $[19, 36]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 26$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(26) + (-27) = -1$, so no completion satisfies the deadline.

15.24 T24. Snapshot boundary, asymmetric delivery

The paired marker observations bracket two channel records, followed by a shared snapshot boundary and an export event.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 4]$	primary
A	C	$[6, 13]$	primary
B	D	$[9, 10]$	primary
C	E	$[5, 12]$	primary
D	F	$[11, 12]$	primary
E	F	$[3, 10]$	primary
F	G	$[4, 5]$	primary
B	C	$[4, 8]$	calibration
D	E	$[5, 5]$	calibration
A	G	$[27, 31]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 4 & 12 & 14 & 19 & 26 & 31 \\ -3 & 0 & 8 & 10 & 15 & 22 & 27 \\ -7 & -4 & 0 & 6 & 11 & 18 & 23 \\ -12 & -9 & -1 & 0 & 5 & 12 & 17 \\ -17 & -14 & -6 & -5 & 0 & 7 & 12 \\ -23 & -20 & -12 & -11 & -6 & 0 & 5 \\ -27 & -24 & -16 & -15 & -10 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 27 to 31. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	3	11	12	17	23	27
Upper	0	4	12	14	19	26	31

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to G$	31	A, G
$G \to A$	-27	G, A

Changed observations Pinning $t_D = 13$ gives endpoint interval $[28, 30]$. Removing

the comparison on B–C with interval $[4, 8]$ gives $[27, 31]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 26$. This contributes an edge from A to the endpoint. Follow it by $G \to A$. The resulting cycle has total weight $(26) + (-27) = -1$, so no completion satisfies the deadline.

15.25 T25. Ring calibration, uniform instrumentation

A serial run is observed from several overlapping time windows. The endpoint calibration closes a cycle in the constraint graph, even though every process dependency runs forward.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 4]$	primary
B	C	$[4, 7]$	primary
C	D	$[5, 8]$	primary
D	E	$[3, 5]$	primary
E	F	$[5, 7]$	primary
A	F	$[24, 26]$	calibration
B	E	$[15, 17]$	calibration
A	D	$[13, 17]$	calibration
C	F	$[16, 18]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 4 & 10 & 17 & 21 & 26 \\ -2 & 0 & 7 & 14 & 17 & 24 \\ -6 & -4 & 0 & 8 & 13 & 18 \\ -13 & -10 & -5 & 0 & 5 & 12 \\ -17 & -15 & -9 & -3 & 0 & 7 \\ -24 & -20 & -16 & -8 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 24 to 26. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	4	8	16	19	24
Upper	0	4	10	17	21	26

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to F$	26	A, F
$F \to A$	-24	F, A

Changed observations Pinning $t_D = 15$ gives endpoint interval $[24, 26]$. Removing the comparison on A–F with interval $[24, 26]$ gives $[22, 28]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 23$. This contributes an edge from A to the endpoint. Follow it by $F \to A$. The resulting cycle has total weight $(23) + (-24) = -1$, so no completion satisfies the deadline.

15.26 T26. Ring calibration, tight cross-window

A serial run is observed from several overlapping time windows. The endpoint calibration closes a cycle in the constraint graph, even though every process dependency runs forward.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[1, 7]$	primary
B	C	$[4, 8]$	primary
C	D	$[0, 6]$	primary
D	E	$[3, 7]$	primary
E	F	$[4, 10]$	primary
A	F	$[25, 25]$	calibration
B	E	$[13, 14]$	calibration
A	D	$[13, 14]$	calibration
C	F	$[15, 15]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 6 & 10 & 14 & 20 & 25 \\ -2 & 0 & 8 & 11 & 14 & 23 \\ -10 & -4 & 0 & 4 & 10 & 15 \\ -13 & -7 & -3 & 0 & 7 & 12 \\ -16 & -13 & -6 & -3 & 0 & 9 \\ -25 & -19 & -15 & -11 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 25 to 25. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	6	10	14	20	25
Upper	0	6	10	14	20	25

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to F$	25	A, F
$F \to A$	-25	F, A

Changed observations Pinning $t_D = 13$ gives endpoint interval $[25, 25]$. Removing the comparison on A–F with interval $[25, 25]$ gives $[22, 29]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 24$. This contributes an edge from A to the endpoint. Follow it by $F \to A$. The resulting cycle has total weight $(24) + (-25) = -1$, so no completion satisfies the deadline.

15.27 T27. Ring calibration, asymmetric delivery

A serial run is observed from several overlapping time windows. The endpoint calibration closes a cycle in the constraint graph, even though every process dependency runs forward.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 8]$	primary
B	C	$[3, 8]$	primary
C	D	$[4, 8]$	primary
D	E	$[3, 6]$	primary
E	F	$[2, 6]$	primary
A	F	$[25, 27]$	calibration
B	E	$[15, 19]$	calibration
A	D	$[14, 16]$	calibration
C	F	$[13, 15]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 12 & 16 & 22 & 27 \\ -4 & 0 & 8 & 12 & 18 & 23 \\ -10 & -3 & 0 & 6 & 12 & 15 \\ -14 & -9 & -4 & 0 & 6 & 11 \\ -19 & -15 & -7 & -3 & 0 & 6 \\ -25 & -18 & -13 & -9 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 25 to 27. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	7	12	16	22	25
Upper	0	7	12	16	22	27

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to F$	27	A, F
$F \to A$	-25	F, A

Changed observations Pinning $t_D = 15$ gives endpoint interval $[25, 26]$. Removing the comparison on A–F with interval $[25, 27]$ gives $[21, 27]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 24$. This contributes an edge from A to the endpoint. Follow it by $F \to A$. The resulting cycle has total weight $(24) + (-25) = -1$, so no completion satisfies the deadline.

15.28 T28. Shared origin, two clocks, uniform instrumentation

Two clock domains agree on an origin and expose local increments. Cross-domain comparisons impose additional bounds on the offset.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 6]$	primary
B	C	$[5, 7]$	primary
A	D	$[11, 14]$	primary
D	E	$[4, 6]$	primary
B	D	$[7, 10]$	calibration
C	E	$[7, 9]$	calibration
B	E	$[13, 16]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 6 & 13 & 14 & 20 \\ -3 & 0 & 7 & 10 & 16 \\ -8 & -5 & 0 & 5 & 9 \\ -11 & -7 & -1 & 0 & 6 \\ -16 & -13 & -7 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 16 to 20. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	3	9	12	16
Upper	0	6	13	14	20

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to D \to E$	20	A, E
$E \to B \to A$	-16	E, A

Changed observations Pinning $t_C = 10$ gives endpoint interval $[17, 19]$. Removing the comparison on B–E with interval $[13, 16]$ gives $[15, 20]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 15$. This contributes an edge from A to the endpoint. Follow it by $E \to B \to A$. The resulting cycle has total weight $(15) + (-16) = -1$, so no completion satisfies the deadline.

15.29 T29. Shared origin, two clocks, tight cross-window

Two clock domains agree on an origin and expose local increments. Cross-domain comparisons impose additional bounds on the offset.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 7]$	primary
B	C	$[5, 10]$	primary
A	D	$[13, 18]$	primary
D	E	$[4, 9]$	primary
B	D	$[11, 11]$	calibration
C	E	$[10, 10]$	calibration
B	E	$[16, 18]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 15 & 18 & 25 \\ -2 & 0 & 8 & 11 & 18 \\ -8 & -6 & 0 & 5 & 10 \\ -13 & -11 & -3 & 0 & 7 \\ -18 & -16 & -10 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 18 to 25. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	2	8	13	18
Upper	0	7	15	18	25

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to B \to E$	25	A, E
$E \to B \to A$	-18	E, A

Changed observations Pinning $t_C = 11$ gives endpoint interval $[21, 21]$. Removing the comparison on B–E with interval $[16, 18]$ gives $[17, 27]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 17$. This contributes an edge from A to the endpoint. Follow it by $E \to B \to A$. The resulting cycle has total weight $(17) + (-18) = -1$, so no completion satisfies the deadline.

15.30 T30. Shared origin, two clocks, asymmetric delivery

Two clock domains agree on an origin and expose local increments. Cross-domain comparisons impose additional bounds on the offset.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[5, 7]$	primary
B	C	$[-1, 7]$	primary
A	D	$[14, 14]$	primary
D	E	$[4, 10]$	primary
B	D	$[6, 10]$	calibration
C	E	$[12, 14]$	calibration
B	E	$[13, 15]$	calibration

Distance closure Rows and columns are indexed by A through E. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 10 & 14 & 22 \\ -5 & 0 & 3 & 9 & 15 \\ -4 & 1 & 0 & 10 & 14 \\ -14 & -7 & -4 & 0 & 8 \\ -18 & -13 & -12 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_E ranges from 18 to 22. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E
Lower	0	5	6	14	18
Upper	0	7	10	14	22

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to B \to E$	22	A, E
$E \to B \to A$	-18	E, A

Changed observations Pinning $t_C = 7$ gives endpoint interval $[19, 21]$. Removing the comparison on B–E with interval $[13, 15]$ gives $[18, 24]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_E \le 17$. This contributes an edge from A to the endpoint. Follow it by $E \to B \to A$. The resulting cycle has total weight $(17) + (-18) = -1$, so no completion satisfies the deadline.

15.31 T31. Quorum collection window, uniform instrumentation

Five reply windows constrain one collection event. The model is conditional on the named replies, so timing feasibility does not count which other reply sets are available.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 6]$	primary
A	C	$[11, 13]$	primary
A	D	$[14, 17]$	primary
A	E	$[21, 24]$	primary
A	F	$[24, 26]$	primary
B	G	$[24, 26]$	primary
C	G	$[16, 19]$	primary
D	G	$[13, 16]$	primary
E	G	$[7, 9]$	primary
F	G	$[4, 6]$	primary
B	D	$[9, 12]$	calibration
C	E	$[9, 12]$	calibration
D	F	$[8, 10]$	calibration
C	G	$[16, 19]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 6 & 13 & 17 & 24 & 26 & 32 \\ -4 & 0 & 9 & 12 & 19 & 22 & 26 \\ -11 & -5 & 0 & 6 & 12 & 15 & 19 \\ -14 & -9 & -1 & 0 & 9 & 10 & 16 \\ -21 & -15 & -9 & -4 & 0 & 5 & 9 \\ -24 & -18 & -11 & -8 & -1 & 0 & 6 \\ -28 & -24 & -16 & -13 & -7 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 28 to 32. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	4	12	15	21	24	28
Upper	0	6	13	17	24	26	32

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to B \to G$	32	A, G
$G \to B \to A$	-28	G, A

Changed observations Pinning $t_D = 15$ gives endpoint interval $[28, 31]$. Removing the comparison on B–D with interval $[9, 12]$ gives $[28, 32]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 27$. This contributes an edge from A to the endpoint. Follow it by $G \to B \to A$. The resulting cycle has total weight $(27) + (-28) = -1$, so no completion satisfies the deadline.

15.32 T32. Quorum collection window, tight cross-window

Five reply windows constrain one collection event. The model is conditional on the named replies, so timing feasibility does not count which other reply sets are available.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	[3, 9]	primary
A	C	[7, 11]	primary
A	D	[11, 17]	primary
A	E	[19, 23]	primary
A	F	[22, 28]	primary
B	G	[23, 27]	primary
C	G	[19, 25]	primary
D	G	[15, 19]	primary
E	G	[7, 13]	primary
F	G	[4, 8]	primary
B	D	[8, 8]	calibration
C	E	[12, 13]	calibration
D	F	[10, 11]	calibration
C	G	[22, 22]	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 9 & 11 & 17 & 23 & 28 & 33 \\ -3 & 0 & 5 & 8 & 18 & 19 & 27 \\ -7 & -1 & 0 & 7 & 13 & 18 & 22 \\ -11 & -8 & -3 & 0 & 10 & 11 & 19 \\ -19 & -13 & -12 & -5 & 0 & 6 & 10 \\ -22 & -18 & -14 & -10 & -1 & 0 & 8 \\ -29 & -23 & -22 & -15 & -9 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 29 to 33. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	6	7	14	20	25	29
Upper	0	9	11	17	23	28	33

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to G$	33	A, G
$G \to C \to A$	-29	G, A

Changed observations Pinning $t_D = 14$ gives endpoint interval $[29, 33]$. Removing the comparison on C–G with interval $[22, 22]$ gives $[26, 36]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 28$. This contributes an edge from A to the endpoint. Follow it by $G \to C \to A$. The resulting cycle has total weight $(28) + (-29) = -1$, so no completion satisfies the deadline.

15.33 T33. Quorum collection window, asymmetric delivery

Five reply windows constrain one collection event. The model is conditional on the named replies, so timing feasibility does not count which other reply sets are available.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	[6, 11]	primary
A	C	[7, 11]	primary
A	D	[17, 20]	primary
A	E	[17, 21]	primary
A	F	[24, 29]	primary
B	G	[21, 25]	primary
C	G	[21, 24]	primary
D	G	[12, 16]	primary
E	G	[11, 16]	primary
F	G	[3, 7]	primary
B	D	[8, 12]	calibration
C	E	[9, 9]	calibration
D	F	[6, 10]	calibration
C	G	[19, 23]	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 11 & 11 & 20 & 20 & 29 & 34 \\ -6 & 0 & 4 & 12 & 13 & 22 & 25 \\ -8 & 2 & 0 & 11 & 9 & 20 & 23 \\ -17 & -8 & -6 & 0 & 3 & 10 & 16 \\ -17 & -7 & -9 & 2 & 0 & 11 & 14 \\ -24 & -14 & -14 & -6 & -5 & 0 & 7 \\ -29 & -21 & -21 & -12 & -12 & -3 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 29 to 34. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	8	8	17	17	26	29
Upper	0	11	11	20	20	29	34

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to C \to G$	34	A, G
$G \to D \to A$	-29	G, A

Changed observations Pinning $t_D = 18$ gives endpoint interval $[30, 34]$. Removing the comparison on C–G with interval $[19, 23]$ gives $[29, 35]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 28$. This contributes an edge from A to the endpoint. Follow it by $G \to D \to A$. The resulting cycle has total weight $(28) + (-29) = -1$, so no completion satisfies the deadline.

15.34 T34. Nested critical paths, uniform instrumentation

Two parallel paths meet at an internal barrier and then split again. The extra intervals constrain transport across a whole barrier.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[5, 7]$	primary
B	D	$[7, 9]$	primary
A	C	$[7, 11]$	primary
C	D	$[4, 6]$	primary
D	E	$[6, 8]$	primary
D	F	$[10, 12]$	primary
E	G	$[8, 12]$	primary
F	G	$[5, 7]$	primary
A	D	$[13, 15]$	calibration
D	G	$[16, 18]$	calibration
C	F	$[14, 18]$	calibration
B	E	$[14, 16]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 11 & 15 & 23 & 27 & 33 \\ -5 & 0 & 5 & 9 & 16 & 21 & 27 \\ -7 & -1 & 0 & 6 & 14 & 18 & 24 \\ -13 & -7 & -4 & 0 & 8 & 12 & 18 \\ -19 & -14 & -10 & -6 & 0 & 6 & 12 \\ -23 & -17 & -14 & -10 & -2 & 0 & 7 \\ -29 & -23 & -20 & -16 & -8 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 29 to 33. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	6	9	13	21	24	29
Upper	0	7	11	15	23	27	33

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to D \to G$	33	A, G
$G \to D \to A$	-29	G, A

Changed observations Pinning $t_D = 14$ gives endpoint interval $[30, 32]$. Removing the comparison on A–D with interval

$[13, 15]$ gives $[28, 34]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 28$. This contributes an edge from A to the endpoint. Follow it by $G \to D \to A$. The resulting cycle has total weight $(28) + (-29) = -1$, so no completion satisfies the deadline.

15.35 T35. Nested critical paths, tight cross-window

Two parallel paths meet at an internal barrier and then split again. The extra intervals constrain transport across a whole barrier.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[4, 9]$	primary
B	D	$[8, 13]$	primary
A	C	$[8, 13]$	primary
C	D	$[4, 9]$	primary
D	E	$[0, 5]$	primary
D	F	$[6, 11]$	primary
E	G	$[9, 14]$	primary
F	G	$[5, 10]$	primary
A	D	$[17, 17]$	calibration
D	G	$[14, 15]$	calibration
C	F	$[14, 15]$	calibration
B	E	$[13, 13]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 9 & 13 & 17 & 22 & 27 & 32 \\ -4 & 0 & 8 & 13 & 13 & 22 & 27 \\ -8 & 1 & 0 & 9 & 14 & 15 & 24 \\ -17 & -8 & -4 & 0 & 5 & 10 & 15 \\ -17 & -13 & -5 & 0 & 0 & 9 & 14 \\ -23 & -14 & -14 & -6 & -1 & 0 & 9 \\ -31 & -22 & -19 & -14 & -9 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 31 to 32. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	9	12	17	22	26	31
Upper	0	9	13	17	22	27	32

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to D \to G$	32	A, G
$G \to D \to A$	-31	G, A

Changed observations Pinning $t_D = 17$ gives endpoint interval $[31, 32]$. Removing the comparison on A–D with interval $[17, 17]$ gives $[27, 36]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 30$. This contributes an edge from A to the endpoint. Follow it by $G \to D \to A$. The resulting cycle has total weight $(30) + (-31) = -1$, so no completion satisfies the deadline.

15.36 T36. Nested critical paths, asymmetric delivery

Two parallel paths meet at an internal barrier and then split again. The extra intervals constrain transport across a whole barrier.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[2, 3]$	primary
B	D	$[8, 15]$	primary
A	C	$[8, 9]$	primary
C	D	$[4, 11]$	primary
D	E	$[3, 4]$	primary
D	F	$[6, 13]$	primary
E	G	$[9, 10]$	primary
F	G	$[0, 7]$	primary
A	D	$[15, 17]$	calibration
D	G	$[11, 15]$	calibration
C	F	$[15, 17]$	calibration
B	E	$[16, 18]$	calibration

Distance closure Rows and columns are indexed by A through G. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 3 & 9 & 17 & 21 & 26 & 31 \\ -2 & 0 & 7 & 15 & 18 & 24 & 28 \\ -8 & -5 & 0 & 9 & 13 & 17 & 23 \\ -15 & -12 & -6 & 0 & 4 & 11 & 14 \\ -18 & -16 & -9 & -3 & 0 & 8 & 10 \\ -23 & -20 & -15 & -6 & -2 & 0 & 7 \\ -27 & -25 & -18 & -12 & -9 & -1 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_G ranges from 27 to 31. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F	G
Lower	0	2	9	15	18	26	27
Upper	0	3	9	17	21	26	31

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to B \to E \to G$	31	A, G
$G \to E \to B \to A$	-27	G, A

Changed observations Pinning $t_D = 16$ gives endpoint interval $[28, 30]$. Removing the comparison on A–D with interval $[15, 17]$ gives $[27, 31]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_G \le 26$. This contributes an edge from A to the endpoint. Follow it by $G \to E \to B \to A$. The resulting cycle has total weight $(26) + (-27) = -1$, so no completion satisfies the deadline.

15.37 T37. Deferred acknowledgment, uniform instrumentation

The sender can receive a local storage result before the remote acknowledgment. Both observations constrain the reported response time.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[6, 8]$	primary
B	C	$[3, 6]$	primary
C	D	$[4, 7]$	primary
B	E	$[12, 14]$	primary
D	F	$[7, 9]$	primary
E	F	$[4, 7]$	primary
A	E	$[19, 22]$	calibration
C	E	$[7, 10]$	calibration
C	F	$[13, 15]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 8 & 14 & 21 & 22 & 29 \\ -6 & 0 & 6 & 13 & 14 & 21 \\ -9 & -3 & 0 & 7 & 10 & 15 \\ -14 & -7 & -4 & 0 & 5 & 9 \\ -19 & -12 & -7 & 0 & 0 & 7 \\ -23 & -16 & -13 & -7 & -4 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 23 to 29. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	7	10	16	19	23
Upper	0	8	14	21	22	29

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to E \to F$	29	A, F
$F \to E \to A$	-23	F, A

Changed observations Pinning $t_D = 17$ gives endpoint interval $[24, 26]$. Removing the comparison on A–E with interval $[19, 22]$ gives $[22, 29]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 22$. This contributes an edge from A to the endpoint. Follow it by $F \to E \to A$. The resulting cycle has total weight $(22) + (-23) = -1$, so no completion satisfies the deadline.

15.38 T38. Deferred acknowledgment, tight cross-window

The sender can receive a local storage result before the remote acknowledgment. Both observations constrain the reported response time.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[0, 6]$	primary
B	C	$[3, 7]$	primary
C	D	$[4, 10]$	primary
B	E	$[14, 18]$	primary
D	F	$[7, 13]$	primary
E	F	$[4, 8]$	primary
A	E	$[18, 20]$	calibration
C	E	$[11, 11]$	calibration
C	F	$[17, 17]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 6 & 9 & 19 & 20 & 26 \\ 0 & 0 & 7 & 17 & 18 & 24 \\ -7 & -3 & 0 & 10 & 11 & 17 \\ -11 & -7 & -4 & 0 & 7 & 13 \\ -18 & -14 & -11 & -1 & 0 & 6 \\ -24 & -20 & -17 & -7 & -6 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 24 to 26. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	4	7	17	18	24
Upper	0	6	9	19	20	26

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to E \to C \to F$	26	A, F
$F \to C \to E \to A$	-24	F, A

Changed observations Pinning $t_D = 15$ gives endpoint interval $[24, 26]$. Removing the comparison on A–E with interval $[18, 20]$ gives $[20, 30]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 23$. This contributes an edge from A to the endpoint. Follow it by $F \to C \to E \to A$. The resulting cycle has total weight $(23) + (-24) = -1$, so no completion satisfies the deadline.

15.39 T39. Deferred acknowledgment, asymmetric delivery

The sender can receive a local storage result before the remote acknowledgment. Both observations constrain the reported response time.

Input intervals A primary interval is part of the base diagram. A calibration interval is an additional observation.

Start	End	Interval	Kind
A	B	$[3, 7]$	primary
B	C	$[2, 7]$	primary
C	D	$[3, 7]$	primary
B	E	$[11, 14]$	primary
D	F	$[11, 15]$	primary
E	F	$[3, 8]$	primary
A	E	$[16, 18]$	calibration
C	E	$[6, 10]$	calibration
C	F	$[15, 17]$	calibration

Distance closure Rows and columns are indexed by A through F. A negative off-diagonal value is allowed. It expresses a lower bound in the opposite direction.

$$D = \begin{bmatrix} 0 & 7 & 11 & 15 & 18 & 26 \\ -3 & 0 & 7 & 11 & 14 & 22 \\ -6 & -2 & 0 & 6 & 10 & 17 \\ -9 & -5 & -3 & 0 & 7 & 14 \\ -16 & -11 & -7 & -3 & 0 & 8 \\ -21 & -17 & -15 & -11 & -5 & 0 \end{bmatrix}$$

Realizing schedules Set $t_A = 0$. The endpoint t_F ranges from 21 to 26. The two rows of schedules below attain these bounds. Every original interval holds in both rows.

Bound	A	B	C	D	E	F
Lower	0	4	6	10	16	21
Upper	0	7	11	15	18	26

Short path certificates The upper path uses forward summation of inequalities; the lower bound comes from a return path.

Path	Weight	Endpoints
$A \to E \to F$	26	A, F
$F \to C \to E \to A$	-21	F, A

Changed observations Pinning $t_D = 12$ gives endpoint interval $[23, 26]$. Removing the comparison on A–E with interval $[16, 18]$ gives $[20, 29]$. Both calculations retain all other input intervals.

Contradiction certificate Impose the additional deadline $t_F \le 20$. This contributes an edge from A to the endpoint. Follow it by $F \to C \to E \to A$. The resulting cycle has total weight $(20) + (-21) = -1$, so no completion satisfies the deadline.

16 Atlas of abelian matching obstructions

The underlying observation complex is a single vertex, named loops, and one face for each row of the integer matrix. The row's entries are the exponent sums in its attaching word. Labels are taken in the stated prime field and the vertex frame is fixed. A pinned loop contributes a known term to the boundary. See Theorem 5.5 and Corollary 5.6.

16.1 A01. Shared comparison over F2, profile 1

The third attaching word has abelian exponent vector equal to the sum of the first two. Its declared boundary value is constrained by the first two values even if every individual equation can be solved.

Input and pinning The field has characteristic 2. Columns are the loops A through C. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C. The adjusted right-hand side is $(0, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(1, 1, 1)$.

Obstruction certificate The covector $(1, 1, 1)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	16	4
6	6	36	6
8	8	64	8
9	9	81	9
12	12	144	12

16.2 A02. Shared comparison over F3, profile 2

The third attaching word has abelian exponent vector equal to the sum of the first two. Its declared boundary value is constrained by the first two values even if every individual equation can be solved.

Input and pinning The field has characteristic 3. Columns are the loops A through C. Pin $A = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C. The adjusted right-hand side is $(0, 0, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(0, 0)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	4
6	1	36	6
8	1	64	8
9	1	81	9
12	1	144	12

16.3 A03. Shared comparison over F3, profile 3

The third attaching word has abelian exponent vector equal to the sum of the first two. Its declared boundary value is constrained by the first two values even if every individual equation can be solved.

Input and pinning The field has characteristic 3. Columns are the loops A through C. Pin $B = 2$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(1, 1, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(2, 2, 1)$ annihilates every free column but takes value 2 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	4
6	1	36	6
8	1	64	8
9	1	81	9
12	1	144	12

16.4 A04. Two parallel patch reports over F3, profile 1

Two patches record the same boundary word. The comparison equations for those patches must agree, while the other constraints couple their local witness variables to two additional loops.

Input and pinning The field has characteristic 3. Columns are the loops A through D. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D. The adjusted right-hand side is $(2, 1, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 2 & 2 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 2 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(2, 1, 2, 1)$.

Obstruction certificate The covector $(2, 1, 0, 0)$ annihilates every free column but takes value 2 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	64	4
6	6	216	6
8	8	512	8
9	9	729	9
12	12	1,728	12

16.5 A05. Two parallel patch reports over F5, profile 2

Two patches record the same boundary word. The comparison equations for those patches must agree, while the other constraints couple their local witness variables to two additional loops.

Input and pinning The field has characteristic 5. Columns are the loops A through D. Pin $B = 2$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C, D. The adjusted right-hand side is $(4, 4, 1, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 4 \\ 0 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(4, 3, 3)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	4
6	1	216	6
8	1	512	8
9	1	729	9
12	1	1,728	12

16.6 A06. Two parallel patch reports over F2, profile 3

Two patches record the same boundary word. The comparison equations for those patches must agree, while the other constraints couple their local witness variables to two additional loops.

Input and pinning The field has characteristic 2. Columns are the loops A through D. Pin $A = 1$, $C = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, D. The adjusted right-hand side is $(1, 0, 0, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(1, 1, 0, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are

$(1,1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	16
6	1	36	36
8	1	64	64
9	1	81	81
12	1	144	144

16.7 A07. Even winding over F5, profile 1

Winding numbers create torsion. The last report ties the two initial winding variables to a third comparison.

Input and pinning The field has characteristic 5. Columns are the loops A through C. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 1 & 1 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 4 \\ 4 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C. The adjusted right-hand side is $(0,4,4)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 4 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 0. A basis for the kernel is the empty basis.

Lift certificate The vector $(0,1,4)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1,2,8)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	8	8	8
6	4	54	4
8	16	32	16
9	1	729	1
12	8	216	8

16.8 A08. Even winding over F7, profile 2

Winding numbers create torsion. The last report ties the two initial winding variables to a third comparison.

Input and pinning The field has characteristic 7. Columns are the loops A through C. Pin $C = 3$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 1 & 1 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 5 \\ 0 \\ 5 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B. The adjusted right-hand side is $(5,0,6)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 6 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(6, 0)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 2)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	2	8	8
6	2	18	12
8	2	32	16
9	1	81	9
12	2	72	24

16.9 A09. Even winding over F5, profile 3

Winding numbers create torsion. The last report ties the two initial winding variables to a third comparison.

Input and pinning The field has characteristic 5. Columns are the loops A through C. Pin $A = 4$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 4 & 0 \\ 1 & 1 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 4 \\ 2 \\ 4 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C. The adjusted right-hand side is $(1, 2, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(1, 0, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 8)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	4	16
6	2	18	12
8	8	8	64
9	1	81	9
12	4	36	48

16.10 A10. Odd winding over F7, profile 1

Two boundaries have degree three in distinct loop variables. Their sum is also reported, and a separate face constrains the shared interface with a degree-six boundary.

Input and pinning The field has characteristic 7. Columns are the loops A through C. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 3 & 3 & 2 \\ 0 & 0 & 6 \end{bmatrix}, \quad c = \begin{bmatrix} 5 \\ 0 \\ 4 \\ 3 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C. The adjusted right-hand side is $(5, 0, 4, 3)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 6 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(6, 6, 1, 0)$ annihilates every free column but takes value 6 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 3, 18)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	2	32	8
6	18	12	108
8	2	256	16
9	27	27	243
12	18	96	216

16.11 A11. Odd winding over F2, profile 2

Two boundaries have degree three in distinct loop variables. Their sum is also reported, and a separate face constrains the shared interface with a degree-six boundary.

Input and pinning The field has characteristic 2. Columns are the loops A through C. Pin $A = 0$. The frame at the vertex is always fixed.

$$M = \left[\begin{array}{ccc} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 3 & 3 & 2 \\ 0 & 0 & 6 \end{array}\right], \quad c = \left[\begin{array}{c} 0 \\ 1 \\ 1 \\ 0 \end{array}\right]$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C. The adjusted right-hand side is $(0, 1, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Lift certificate The vector $(1, 0)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 3)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	16
6	3	12	108
8	1	64	64
9	3	27	243
12	3	48	432

16.12 A12. Odd winding over F2, profile 3

Two boundaries have degree three in distinct loop variables. Their sum is also reported, and a separate face constrains the shared interface with a degree-six boundary.

Input and pinning The field has characteristic 2. Columns are the loops A through

C. Pin $B = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 3 & 0 & 1 \\ 0 & 3 & 1 \\ 3 & 3 & 2 \\ 0 & 0 & 6 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(1, 1, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(1, 1, 1, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 3)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	16
6	3	12	108
8	1	64	64
9	3	27	243
12	3	48	432

16.13 A13. Pinned offset chain over F2, profile 1

Successive frames differ by reported offsets. The direct endpoint offset must be the telescoping sum of the three differences.

Input and pinning The field has characteristic 2. Columns are the loops A through D. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D. The adjusted right-hand side is $(0, 1, 1, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(1, 1, 1, 1)$.

Obstruction certificate The covector $(1, 1, 1, 1)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	64	4
6	6	216	6
8	8	512	8
9	9	729	9
12	12	1,728	12

16.14 A14. Pinned offset chain over F3, profile 2

Successive frames differ by reported offsets. The direct endpoint offset must be the telescoping sum of the three differences.

Input and pinning The field has characteristic 3. Columns are the loops A through D. Pin $A = 2$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C, D. The adjusted right-hand side is $(1, 0, 1, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(2, 2, 1)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	4
6	1	216	6
8	1	512	8
9	1	729	9
12	1	1,728	12

16.15 A15. Pinned offset chain over F3, profile 3

Successive frames differ by reported offsets. The direct endpoint offset must be the telescoping sum of the three differences.

Input and pinning The field has characteristic 3. Columns are the loops A through D. Pin $B = 0$, $D = 2$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(0, 0, 0, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(2, 0, 0, 1)$ annihilates every free column but takes value 2 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the

homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	16
6	1	36	36
8	1	64	64
9	1	81	81
12	1	144	144

16.16 A16. Two-dimensional comparator over F3, profile 1

Two staged relations are compared with an aggregate relation. The fourth row constrains a degree-two wrap around the endpoints.

Input and pinning The field has characteristic 3. Columns are the loops A through E. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & -1 \\ 2 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D, E. The adjusted right-hand side is $(0, 2, 1, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 1 & 2 \\ 0 & 1 & 0 & 1 & 1 & 2 \\ 0 & 0 & 1 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 2, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(0, 2, 2, 1, 0)$, $(2, 2, 1, 0, 1)$.

Obstruction certificate The covector $(1, 1, 2, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 2)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	32	32	8
6	72	108	12
8	128	256	16
9	81	729	9
12	288	864	24

16.17 A17. Two-dimensional comparator over F5, profile 2

Two staged relations are compared with an aggregate relation. The fourth row constrains a degree-two wrap around the endpoints.

Input and pinning The field has characteristic 5. Columns are the loops A through E. Pin $A = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & -1 \\ 2 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 4 \\ 2 \\ 3 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C, D, E. The adjusted right-hand side is $(2, 4, 1, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 1 & 0 & 4 \\ 0 & 1 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(4, 4, 1, 0)$.

Lift certificate The vector $(4, 2, 0, 3)$ satisfies the relative equation. Adding all kernel vectors gives exactly 5 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 2)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	8	32	8
6	12	108	12
8	16	256	16
9	9	729	9
12	24	864	24

16.18 A18. Two-dimensional comparator over F2, profile 3

Two staged relations are compared with an aggregate relation. The fourth row constrains a degree-two wrap around the endpoints.

Input and pinning The field has characteristic 2. Columns are the loops A through E. Pin $B = 1$, $D = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 1 & -1 \\ 1 & 1 & 0 & 1 & -1 \\ 2 & 0 & 0 & 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C, E. The adjusted right-hand side is $(1, 0, 0, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is $(1, 1, 1)$.

Obstruction certificate The covector $(1, 1, 1, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 4)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	16	16
6	2	108	12
8	4	128	32
9	1	729	9
12	4	432	48

16.19 A19. Coupled loop returns over F5, profile 1

Every equation constrains two successive loop variables, with the last one closing the chain. The integral determinant records when that closure can become singular modulo a prime.

Input and pinning The field has characteristic 5. Columns are the loops A through D. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D. The adjusted right-hand side is $(1, 0, 1, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 2 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 3 & 3 \\ 0 & 0 & 0 & 0 & 2 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(3, 4, 2, 1)$.

Obstruction certificate The covector $(2, 4, 3, 1)$ annihilates every free column but takes value 2 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1, 15)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	256	1
6	3	432	3
8	1	4,096	1
9	3	2,187	3
12	3	6,912	3

16.20 A20. Coupled loop returns over F7, profile 2

Every equation constrains two successive loop variables, with the last one closing the chain. The integral determinant records when that closure can become singular modulo a prime.

Input and pinning The field has characteristic 7. Columns are the loops A through D. Pin $C = 0$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 4 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, D. The adjusted right-hand side is $(1, 2, 2, 4)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(0, 1, 2)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	4
6	1	216	6
8	1	512	8
9	1	729	9
12	1	1,728	12

16.21 A21. Coupled loop returns over F5, profile 3

Every equation constrains two successive loop variables, with the last one closing the chain. The integral determinant records when that closure can become singular modulo a prime.

Input and pinning The field has characteristic 5. Columns are the loops A through D. Pin $B = 0$, $D = 3$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 2 & 1 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 4 \\ 1 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(1, 4, 3, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 3 \\ 0 & 1 & 4 \\ 0 & 0 & 0 \\ 0 & 0 & 2 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(2, 0, 0, 1)$ annihilates every free column but takes value 2 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	16
6	1	36	36
8	1	64	64
9	1	81	81
12	1	144	144

16.22 A22. Redundant incident faces over F7, profile 1

The first four rows are the unsigned incidences of a four-cycle. The fifth is a joint checksum. Separate compatibility of each pair does not ensure simultaneous compatibility.

Input and pinning The field has characteristic 7. Columns are the loops A through D. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 4 \\ 0 \\ 1 \\ 4 \\ 4 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D. The adjusted right-hand side is $(4, 0, 1, 4, 4)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 1 & 5 \\ 0 & 1 & 0 & 6 & 6 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 0 & 0 & 6 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is $(6, 1, 6, 1)$.

Obstruction certificate The covector $(6, 1, 6, 1, 0)$ annihilates every free column but takes value 6 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	64	16
6	6	216	36
8	8	512	64
9	9	729	81
12	12	1,728	144

16.23 A23. Redundant incident faces over F2, profile 2

The first four rows are the unsigned incidences of a four-cycle. The fifth is a joint checksum. Separate compatibility of each pair does not ensure simultaneous compatibility.

Input and pinning The field has characteristic 2. Columns are the loops A through D. Pin $D = 0$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C. The adjusted right-hand side is $(1, 1, 0, 0, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Lift certificate The vector $(0, 1, 0)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	16
6	1	216	36
8	1	512	64
9	1	729	81
12	1	1,728	144

16.24 A24. Redundant incident faces over F2, profile 3

The first four rows are the unsigned incidences of a four-cycle. The fifth is a joint checksum. Separate compatibility of each pair does not ensure simultaneous compatibility.

Input and pinning The field has characteristic 2. Columns are the loops A through D. Pin $A = 1$, $C = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, D. The adjusted right-hand side is $(1, 0, 0, 0, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 3. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(1, 1, 0, 0, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	64
6	1	36	216
8	1	64	512
9	1	81	729
12	1	144	1,728

16.25 A25. Parallel epoch frames over F2, profile 1

Loop labels are compared across five epochs. An explicit closing comparison imposes a compatibility condition independent of whether the first epoch is pinned.

Input and pinning The field has characteristic 2. Columns are the loops A through E. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D, E. The adjusted right-hand side is $(0, 1, 1, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 4. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(1, 1, 1, 1, 1)$.

Obstruction certificate The covector $(1, 1, 1, 1, 1)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	256	4
6	6	1,296	6
8	8	4,096	8
9	9	6,561	9
12	12	20,736	12

16.26 A26. Parallel epoch frames over F3, profile 2

Loop labels are compared across five epochs. An explicit closing comparison imposes a compatibility condition independent of whether the first epoch is pinned.

Input and pinning The field has characteristic 3. Columns are the loops A through E. Pin $D = 0$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 0 \\ 1 \\ 2 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, E. The adjusted right-hand side is $(2, 0, 1, 2, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 4. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(0, 1, 1, 1)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	256	4
6	1	1,296	6
8	1	4,096	8
9	1	6,561	9
12	1	20,736	12

16.27 A27. Parallel epoch frames over F3, profile 3

Loop labels are compared across five epochs. An explicit closing comparison imposes a compatibility condition independent of whether the first epoch is pinned.

Input and pinning The field has characteristic 3. Columns are the loops A through E. Pin $B = 0$, $E = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 1 & 0 & 0 & 0 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 1 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C, D. The adjusted right-hand side is $(2, 1, 1, 1, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(2, 0, 0, 0, 1)$ annihilates every free column but takes value 2 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	16
6	1	216	36
8	1	512	64
9	1	729	81
12	1	1,728	144

16.28 A28. Skew return map over F3, profile 1

The first two reports couple their coordinates symmetrically, while the third is asymmetric. Fixing one shared coordinate can change the obstruction independently of the determinant.

Input and pinning The field has characteristic 3. Columns are the loops A through C. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 3 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C. The adjusted right-hand side is $(2, 2, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(0, 0, 1)$.

Obstruction certificate The covector $(1, 1, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 9)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	1
6	3	72	3
8	1	512	1
9	9	81	9
12	3	576	3

16.29 A29. Skew return map over F5, profile 2

The first two reports couple their coordinates symmetrically, while the third is asymmetric. Fixing one shared coordinate can change the obstruction independently of the determinant.

Input and pinning The field has characteristic 5. Columns are the loops A through C. Pin $A = 0$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 3 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C. The adjusted right-hand side is $(1, 2, 3)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 4 \\ 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(1, 4)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 3)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	4
6	3	12	18
8	1	64	8
9	3	27	27
12	3	48	36

16.30 A30. Skew return map over F2, profile 3

The first two reports couple their coordinates symmetrically, while the third is asymmetric. Fixing one shared coordinate can change the obstruction independently of the determinant.

Input and pinning The field has characteristic 2. Columns are the loops A through C. Pin $B = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 2 & 0 \\ 0 & 1 & 3 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(1, 1, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(1, 0, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 3)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	4
6	3	12	18
8	1	64	8
9	3	27	27
12	3	48	36

16.31 A31. Two paths and a checksum over F5, profile 1

The first two faces measure independent paths. The third measures their difference and the fourth twice their sum.

Input and pinning The field has characteristic 5. Columns are the loops A through D. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 4 \\ 0 \\ 3 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D. The adjusted right-hand side is $(4, 0, 3, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 3 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 2, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is $(4, 1, 0, 0)$, $(0, 0, 4, 1)$.

Obstruction certificate The covector $(4, 1, 1, 0)$ annihilates every free column but takes value 4 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	16	16	16
6	36	36	36
8	64	64	64
9	81	81	81
12	144	144	144

16.32 A32. Two paths and a checksum over F7, profile 2

The first two faces measure independent paths. The third measures their difference and the fourth twice their sum.

Input and pinning The field has characteristic 7. Columns are the loops A through D. Pin $C = 4$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 0 \\ 3 \\ 6 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, D. The adjusted right-hand side is $(3, 3, 0, 5)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 1 & 0 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is $(6, 1, 0)$.

Lift certificate The vector $(3, 0, 3)$ satisfies the relative equation. Adding all kernel vectors gives exactly 7 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	16	16
6	6	36	36
8	8	64	64
9	9	81	81
12	12	144	144

16.33 A33. Two paths and a checksum over F5, profile 3

The first two faces measure independent paths. The third measures their difference

and the fourth twice their sum.

Input and pinning The field has characteristic 5. Columns are the loops A through D. Pin $B = 4$, $D = 2$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 2 & 2 & 2 & 2 \end{bmatrix}, \quad c = \begin{bmatrix} 3 \\ 4 \\ 3 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(4, 2, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 1 & 0 & 4 \\ 0 & 1 & 2 \\ 0 & 0 & 4 \\ 0 & 0 & 3 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(4, 1, 1, 0)$ annihilates every free column but takes value 4 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	16	16
6	1	36	36
8	1	64	64
9	1	81	81
12	1	144	144

16.34 A34. Mixed torsion blocks over F7, profile 1

One coefficient block closes separately. The other couples two winding relations through a common frame variable.

Input and pinning The field has characteristic 7. Columns are the loops A through E. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2 & 1 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 1 \\ 0 & 0 & 0 & 5 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 2 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D, E. The adjusted right-hand side is $(0, 2, 2, 3, 4)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 1 & 0 & 0 & 6 \\ 0 & 0 & 0 & 1 & 0 & 4 \\ 0 & 0 & 0 & 0 & 1 & 4 \end{array}\right]$$

Kernel and obstruction The pivot count is 5. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 0. A basis for the kernel is the empty basis.

Lift certificate The vector $(2, 3, 6, 4, 4)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1, 2, 30)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	256	4
6	12	648	12
8	4	8,192	4
9	3	19,683	3
12	12	20,736	12

16.35 A35. Mixed torsion blocks over F2, profile 2

One coefficient block closes separately. The other couples two winding relations through a common frame variable.

Input and pinning The field has characteristic 2. Columns are the loops A through E. Pin $B = 0$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2&1&0&0&0\\ 0&3&0&0&0\\ 0&0&2&0&1\\ 0&0&0&5&1\\ 0&0&0&0&1 \end{bmatrix}, \quad c = \begin{bmatrix} 0\\0\\0\\1\\0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C, D, E. The adjusted right-hand side is $(0, 0, 0, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 0&0&1&0&1\\ 0&0&0&1&0\\ 0&0&0&0&0\\ 0&0&0&0&0\\ 0&0&0&0&0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 2, and the quotient of required face values by the realizable ones has dimension 3. A basis for the kernel is $(1, 0, 0, 0)$, $(0, 1, 0, 0)$.

Lift certificate The vector $(0, 0, 1, 0)$ satisfies the relative equation. Adding all kernel vectors gives exactly 4 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 2, 10)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	64	16
6	4	324	24
8	4	1,024	32
9	1	6,561	9
12	4	5,184	48

16.36 A36. Mixed torsion blocks over F2, profile 3

One coefficient block closes separately. The other couples two winding relations through a common frame variable.

Input and pinning The field has characteristic 2. Columns are the loops A through E. Pin $C = 1$, $E = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 2&1&0&0&0\\ 0&3&0&0&0\\ 0&0&2&0&1\\ 0&0&0&5&1\\ 0&0&0&0&1 \end{bmatrix}, \quad c = \begin{bmatrix} 1\\0\\1\\1\\1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, D. The adjusted right-hand side is $(1, 0, 0, 0, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 0&1&0&1\\ 0&0&1&0\\ 0&0&0&0\\ 0&0&0&1\\ 0&0&0&0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 3. A basis for the kernel is $(1, 0, 0)$.

Obstruction certificate The covector $(1, 1, 0, 0, 0)$ annihilates every free column

but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 30)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	2	32	32
6	6	36	216
8	2	256	128
9	3	243	243
12	6	288	864

16.37 A37. Free relabeling and one measured loop over F2, profile 1

One loop is entirely unmeasured. Two records of the second loop constrain its winding, and the last report sees only the difference of the two remaining loops.

Input and pinning The field has characteristic 2. Columns are the loops A through D. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D. The adjusted right-hand side is $(1, 0, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 1. The kernel dimension is 3, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is $(1, 0, 0, 0)$, $(0, 1, 0, 0)$, $(0, 0, 1, 1)$.

Obstruction certificate The covector $(1, 0, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 2)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	32	8	8
6	72	18	12
8	128	32	16
9	81	81	9
12	288	72	24

16.38 A38. Free relabeling and one measured loop over F3, profile 2

One loop is entirely unmeasured. Two records of the second loop constrain its winding, and the last report sees only the difference of the two remaining loops.

Input and pinning The field has characteristic 3. Columns are the loops A through D. Pin $A = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are B, C, D. The adjusted right-hand side is $(0, 0, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 2. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(0, 1, 1)$.

Lift certificate The vector $(0, 1, 0)$ satisfies the relative equation. Adding all kernel vectors gives exactly 3 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 2)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	8	8	8
6	12	18	12
8	16	32	16
9	9	81	9
12	24	72	24

16.39 A39. Free relabeling and one measured loop over F3, profile 3

One loop is entirely unmeasured. Two records of the second loop constrain its winding, and the last report sees only the difference of the two remaining loops.

Input and pinning The field has characteristic 3. Columns are the loops A through D. Pin $B = 2$, $D = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 0 & 2 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C. The adjusted right-hand side is $(1, 0, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cc|c} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 1. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is $(1, 0)$.

Obstruction certificate The covector $(1, 0, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are (1). The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	4	16
6	6	6	36
8	8	8	64
9	9	9	81
12	12	12	144

16.40 A40. Branch consistency over F3, profile 1

Two branches from one base comparison meet through an endpoint constraint. Two root-to-endpoint sums must then agree.

Input and pinning The field has characteristic 3. Columns are the loops A through E. No loop is pinned. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 0 \\ 2 \\ 1 \\ 0 \\ 2 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, D, E. The adjusted right-hand side is $(0, 2, 1, 0, 2)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccccc|c} 1 & 0 & 0 & 0 & 2 & 2 \\ 0 & 1 & 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 1 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{array}\right]$$

Kernel and obstruction The pivot count is 4. The kernel dimension is 1, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is $(1, 1, 1, 1, 1)$.

Obstruction certificate The covector $(1, 2, 1, 2, 1)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	4	256	4
6	6	1,296	6
8	8	4,096	8
9	9	6,561	9
12	12	20,736	12

16.41 A41. Branch consistency over F5, profile 2

Two branches from one base comparison meet through an endpoint constraint. Two root-to-endpoint sums must then agree.

Input and pinning The field has characteristic 5. Columns are the loops A through E. Pin $D = 4$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 4 \\ 1 \\ 0 \\ 3 \\ 0 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, B, C, E. The adjusted right-hand side is $(4, 1, 4, 3, 1)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 3 \\ 0 & 1 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & 2 \\ 0 & 0 & 0 & 1 & 4 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 4. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 1. A basis for the kernel is the empty basis.

Lift certificate The vector $(3, 4, 2, 4)$ satisfies the relative equation. Adding all kernel vectors gives exactly 1 lifts. The fixed frame removes degree-zero automorphisms, so this is also the number of components.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	256	4
6	1	1,296	6
8	1	4,096	8
9	1	6,561	9
12	1	20,736	12

16.42 A42. Branch consistency over F2, profile 3

Two branches from one base comparison meet through an endpoint constraint. Two root-to-endpoint sums must then agree.

Input and pinning The field has characteristic 2. Columns are the loops A through E. Pin $B = 1$, $E = 1$. The frame at the vertex is always fixed.

$$M = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}, \quad c = \begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \\ 1 \end{bmatrix}$$

Relative linear equation Delete pinned columns and subtract their contribution from the right-hand side. The remaining columns in order are A, C, D. The adjusted right-hand side is $(0, 0, 0, 1, 0)$.

$$[LM_{\mathrm{free}} \mid Lc_{\mathrm{adj}}] = \left[\begin{array}{ccc|c} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{array}\right]$$

Kernel and obstruction The pivot count is 3. The kernel dimension is 0, and the quotient of required face values by the realizable ones has dimension 2. A basis for the kernel is the empty basis.

Obstruction certificate The covector $(1, 1, 0, 1, 0)$ annihilates every free column but takes value 1 on the required relative boundary. Hence the completion space is empty.

Changing coefficients The nonzero Smith factors of the relative integer matrix are $(1, 1, 1)$. The table compares the kernel of the homogeneous equation and the size of the image and cokernel over cyclic coefficient groups. It keeps the pinning pattern but replaces the right-hand side by zero for the kernel calculation.

Modulus	Kernel	Image	Cokernel
4	1	64	16
6	1	216	36
8	1	512	64
9	1	729	81
12	1	1,728	144

17 Atlas of finite matching groupoids

Each case gives a presentation of the observation complex and a finite target group. C_m is cyclic of order m. $D_{2m} = \langle r, s \mid r^m = s^2 = 1,\ srs = r^{-1} \rangle$ has order $2m$. S_3 and A_4 act on the indicated sets by permutations. $Q_8 = \{\pm 1, \pm i, \pm j, \pm k\}$ is the quaternion group. Framing keeps the actual labels. The unframed groupoid divides by simultaneous conjugation as in Theorem 5.1.

17.1 G01. Single return in C5

The interface graph has one independent cycle and no relation. A framed matching assigns one holonomy. Dropping the frame compares assignments by conjugation.

Presentation Generators are a, with no relations. The target is C_5 of order 5. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	5	5
r	1	5	5
r^2	1	5	5
r^3	1	5	5
r^4	1	5	5

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
r	1	1
r^2	1	1
r^3	1	1
r^4	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
5	5	(1)

Comparison There are 5 framed tuples and 5 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at 1 leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.2 G02. Single return in S3

The interface graph has one independent cycle and no relation. A framed matching assigns one holonomy. Dropping the frame compares assignments by conjugation.

Presentation Generators are a, with no relations. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	6
(2 3)	3	2	2
(1 2 3)	2	3	3

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3)	1	1
(1 2 3)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	1	((2 3))
3	1	((1 2 3))
6	1	(1)

Comparison There are 6 framed tuples and 3 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at (2 3) leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.3 G03. Single return in D8

The interface graph has one independent cycle and no relation. A framed matching assigns one holonomy. Dropping the frame compares assignments by conjugation.

Presentation Generators are a, with no relations. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	8
s	2	4	4
r	2	4	4
rs	2	4	4
r^2	1	8	8

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
s	1	1
r	1	1
rs	1	1
r^2	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	3	(s)
8	2	(1)

Comparison There are 8 framed tuples and 5 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at s leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.4 G04. Single return in Q8

The interface graph has one independent cycle and no relation. A framed matching assigns one holonomy. Dropping the frame compares assignments by conjugation.

Presentation Generators are a, with no relations. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	8
i	2	4	4
j	2	4	4
k	2	4	4
-1	1	8	8

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
i	1	1
j	1	1
k	1	1
-1	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	3	(i)
8	2	(1)

Comparison There are 8 framed tuples and 5 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at i leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.5 G05. Single return in A4

The interface graph has one independent cycle and no relation. A framed matching assigns one holonomy. Dropping the frame compares assignments by conjugation.

Presentation Generators are a, with no relations. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	12
(2 3 4)	4	3	3
(2 4 3)	4	3	3
(1 2)(3 4)	3	4	4

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3 4)	1	1
(2 4 3)	1	1
(1 2)(3 4)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
3	2	((2 3 4))
4	1	((1 2)(3 4))
12	1	(1)

Comparison There are 12 framed tuples and 4 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at (2 3 4) leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.6 G06. Two free returns in C6

Two returns share a single reference frame. Each return is unconstrained, but the same change of frame acts on both labels.

Presentation Generators are a, b, with no relations. The target is C_6 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	36
r	1	6	36
r^2	1	6	36
r^3	1	6	36
r^4	1	6	36
r^5	1	6	36

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	6	6
r	6	6
r^2	6	6
r^3	6	6
r^4	6	6
r^5	6	6

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
6	36	$(1, 1)$

Comparison There are 36 framed tuples and 36 unframed components. The groupoid cardinality is 6. Fixing the first holonomy at 1 leaves 6 framed tuples, whereas fixing it at the identity leaves 6.

17.7 G07. Two free returns in S3

Two returns share a single reference frame. Each return is unconstrained, but the same change of frame acts on both labels.

Presentation Generators are a, b, with no relations. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	36
$(2\,3)$	3	2	4
$(1\,2\,3)$	2	3	9

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	6	3
$(2\,3)$	6	4
$(1\,2\,3)$	6	4

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	3	$((2\,3), (1\,2))$
2	3	$(1, (2\,3))$
3	4	$(1, (1\,2\,3))$
6	1	$(1, 1)$

Comparison There are 36 framed tuples and 11 unframed components. The groupoid cardinality is 6. Fixing the first holonomy at $(2\,3)$ leaves 6 framed tuples, whereas fixing it at the identity leaves 6.

17.8 G08. Two free returns in D8

Two returns share a single reference frame. Each return is unconstrained, but the same change of frame acts on both labels.

Presentation Generators are a, b, with no relations. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	64
s	2	4	16
r	2	4	16
rs	2	4	16
r^2	1	8	64

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	8	5
s	8	6
r	8	6
rs	8	6
r^2	8	5

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	6	(s, r)
4	18	$(1, s)$
8	4	$(1, 1)$

Comparison There are 64 framed tuples and 28 unframed components. The groupoid cardinality is 8. Fixing the first holonomy at s leaves 8 framed tuples, whereas fixing it at the identity leaves 8.

17.9 G09. Two free returns in Q8

Two returns share a single reference frame. Each return is unconstrained, but the same change of frame acts on both labels.

Presentation Generators are a, b, with no relations. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	64
i	2	4	16
j	2	4	16
k	2	4	16
-1	1	8	64

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	8	5
i	8	6
j	8	6
k	8	6
-1	8	5

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	6	(i, j)
4	18	$(1, i)$
8	4	$(1, 1)$

Comparison There are 64 framed tuples and 28 unframed components. The groupoid cardinality is 8. Fixing the first holonomy at i leaves 8 framed tuples, whereas fixing it at the identity leaves 8.

17.10 G10. Two free returns in D10

Two returns share a single reference frame. Each return is unconstrained, but the same

change of frame acts on both labels.

Presentation Generators are a, b, with no relations. The target is D_{10} of order 10. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	10	100
s	5	2	4
r	2	5	25
r^2	2	5	25

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	10	4
s	10	6
r	10	6
r^2	10	6

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	6	(s, r)
2	3	$(1, s)$
5	12	$(1, r)$
10	1	$(1, 1)$

Comparison There are 100 framed tuples and 22 unframed components. The groupoid cardinality is 10. Fixing the first holonomy at s leaves 10 framed tuples, whereas fixing it at the identity leaves 10.

17.11 G11. Two free returns in A4

Two returns share a single reference frame. Each return is unconstrained, but the same change of frame acts on both labels.

Presentation Generators are a, b, with no relations. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	144
(2 3 4)	4	3	9
(2 4 3)	4	3	9
(1 2)(3 4)	3	4	16

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	12	4
(2 3 4)	12	6
(2 4 3)	12	6
(1 2)(3 4)	12	6

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	8	$((2\,3\,4),(1\,2)(3\,4))$
3	8	$(1,(2\,3\,4))$
4	5	$(1,(1\,2)(3\,4))$
12	1	$(1,1)$

Comparison There are 144 framed tuples and 22 unframed components. The groupoid cardinality is 12. Fixing the first holonomy at $(2\,3\,4)$ leaves 12 framed tuples, whereas fixing it at the identity leaves 12.

17.12 G12. Three free returns in S3

The shared frame acts simultaneously on three independent holonomies. Independent conjugation of the three coordinates would change the matching groupoid.

Presentation Generators are a, b, c, with no relations. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	216
$(2\,3)$	3	2	8
$(1\,2\,3)$	2	3	27

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	36	11
$(2\,3)$	36	20
$(1\,2\,3)$	36	18

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	28	$(1,(2\,3),(1\,2))$
2	7	$(1,1,(2\,3))$
3	13	$(1,1,(1\,2\,3))$
6	1	$(1,1,1)$

Comparison There are 216 framed tuples and 49 unframed components. The groupoid cardinality is 36. Fixing the first holonomy at $(2\,3)$ leaves 36 framed tuples, whereas fixing it at the identity leaves 36.

17.13 G13. Three free returns in D8

The shared frame acts simultaneously on three independent holonomies. Independent conjugation of the three coordinates would change the matching groupoid.

Presentation Generators are a, b, c, with no relations. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	512
s	2	4	64
r	2	4	64
rs	2	4	64
r^2	1	8	512

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last

column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	64	28
s	64	40
r	64	40
rs	64	40
r^2	64	28

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	84	$(1, s, r)$
4	84	$(1, 1, s)$
8	8	$(1, 1, 1)$

Comparison There are 512 framed tuples and 176 unframed components. The groupoid cardinality is 64. Fixing the first holonomy at s leaves 64 framed tuples, whereas fixing it at the identity leaves 64.

17.14 G14. Three free returns in Q8

The shared frame acts simultaneously on three independent holonomies. Independent conjugation of the three coordinates would change the matching groupoid.

Presentation Generators are a, b, c, with no relations. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	512
i	2	4	64
j	2	4	64
k	2	4	64
-1	1	8	512

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	64	28
i	64	40
j	64	40
k	64	40
-1	64	28

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	84	$(1, i, j)$
4	84	$(1, 1, i)$
8	8	$(1, 1, 1)$

Comparison There are 512 framed tuples and 176 unframed components. The groupoid cardinality is 64. Fixing the first holonomy at i leaves 64 framed tuples, whereas fixing it at the identity leaves 64.

17.15 G15. Three free returns in A4

The shared frame acts simultaneously on three independent holonomies. Independent conjugation of the three coordinates would change the matching groupoid.

Presentation Generators are a, b, c, with no relations. The target is A_4 of order

12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	1,728
(2 3 4)	4	3	27
(2 4 3)	4	3	27
(1 2)(3 4)	3	4	64

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	144	22
(2 3 4)	144	54
(2 4 3)	144	54
(1 2)(3 4)	144	48

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	130	$(1, (2\,3\,4), (1\,2)(3\,4))$
3	26	$(1, 1, (2\,3\,4))$
4	21	$(1, 1, (1\,2)(3\,4))$
12	1	$(1, 1, 1)$

Comparison There are 1,728 framed tuples and 178 unframed components. The groupoid cardinality is 144. Fixing the first holonomy at (2 3 4) leaves 144 framed tuples, whereas fixing it at the identity leaves 144.

17.16 G16. Commuting two-face in C6

A filled comparison square identifies the two orders of transport. The equation forces the holonomies to commute. A prescribed first label therefore restricts the second to its centralizer.

Presentation Generators are a, b, with $aba^{-1}b^{-1} = 1$. The target is C_6 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	36
r	1	6	36
r^2	1	6	36
r^3	1	6	36
r^4	1	6	36
r^5	1	6	36

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	6	6
r	6	6
r^2	6	6
r^3	6	6
r^4	6	6
r^5	6	6

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
6	36	$(1, 1)$

Comparison There are 36 framed tuples and 36 unframed components. The groupoid cardinality is 6. Fixing the first holonomy at 1 leaves 6 framed tuples, whereas fixing it at the identity leaves 6.

17.17 G17. Commuting two-face in S3

A filled comparison square identifies the two orders of transport. The equation forces the holonomies to commute. A prescribed first label therefore restricts the second to its centralizer.

Presentation Generators are a, b, with $aba^{-1}b^{-1} = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	18
(2 3)	3	2	4
(1 2 3)	2	3	9

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	6	3
(2 3)	2	2
(1 2 3)	3	3

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	3	$(1, (2\,3))$
3	4	$(1, (1\,2\,3))$
6	1	$(1, 1)$

Comparison There are 18 framed tuples and 8 unframed components. The groupoid cardinality is 3. Fixing the first holonomy at (2 3) leaves 2 framed tuples, whereas fixing it at the identity leaves 6.

17.18 G18. Commuting two-face in D8

A filled comparison square identifies the two orders of transport. The equation forces the holonomies to commute. A prescribed first label therefore restricts the second to its centralizer.

Presentation Generators are a, b, with $aba^{-1}b^{-1} = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	40
s	2	4	16
r	2	4	16
rs	2	4	16
r^2	1	8	40

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last

column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	8	5
s	4	4
r	4	4
rs	4	4
r^2	8	5

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	18	$(1, s)$
8	4	$(1, 1)$

Comparison There are 40 framed tuples and 22 unframed components. The groupoid cardinality is 5. Fixing the first holonomy at s leaves 4 framed tuples, whereas fixing it at the identity leaves 8.

17.19 G19. Commuting two-face in Q8

A filled comparison square identifies the two orders of transport. The equation forces the holonomies to commute. A prescribed first label therefore restricts the second to its centralizer.

Presentation Generators are a, b, with $aba^{-1}b^{-1} = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	40
i	2	4	16
j	2	4	16
k	2	4	16
-1	1	8	40

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	8	5
i	4	4
j	4	4
k	4	4
-1	8	5

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	18	$(1, i)$
8	4	$(1, 1)$

Comparison There are 40 framed tuples and 22 unframed components. The groupoid cardinality is 5. Fixing the first holonomy at i leaves 4 framed tuples, whereas fixing it at the identity leaves 8.

17.20 G20. Commuting two-face in D10

A filled comparison square identifies the two orders of transport. The equation forces the holonomies to commute. A prescribed first label therefore restricts the second to its centralizer.

Presentation Generators are a, b, with $aba^{-1}b^{-1} = 1$. The target is D_{10} of order

10. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	10	40
s	5	2	4
r	2	5	25
r^2	2	5	25

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	10	4
s	2	2
r	5	5
r^2	5	5

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	3	$(1, s)$
5	12	$(1, r)$
10	1	$(1, 1)$

Comparison There are 40 framed tuples and 16 unframed components. The groupoid cardinality is 4. Fixing the first holonomy at s leaves 2 framed tuples, whereas fixing it at the identity leaves 10.

17.21 G21. Commuting two-face in A4

A filled comparison square identifies the two orders of transport. The equation forces the holonomies to commute. A prescribed first label therefore restricts the second to its centralizer.

Presentation Generators are a, b, with $aba^{-1}b^{-1} = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	48
$(2\,3\,4)$	4	3	9
$(2\,4\,3)$	4	3	9
$(1\,2)(3\,4)$	3	4	16

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	12	4
$(2\,3\,4)$	3	3
$(2\,4\,3)$	3	3
$(1\,2)(3\,4)$	4	4

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
3	8	$(1, (2\,3\,4))$
4	5	$(1, (1\,2)(3\,4))$
12	1	$(1, 1)$

Comparison There are 48 framed tuples and 14 unframed components. The groupoid cardinality is 4. Fixing the first holonomy at $(2\,3\,4)$ leaves 3 framed tuples, whereas fixing it at the identity leaves 12.

17.22 G22. Three commuting returns in S3

Three 2-cells impose pairwise commutation. In a 1-type no further choice of 3-cell is required to classify the matching groupoid.

Presentation Generators are a, b, c, with $aba^{-1}b^{-1} = 1$, $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	48
$(2\,3)$	3	2	8
$(1\,2\,3)$	2	3	27

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	18	8
$(2\,3)$	4	4
$(1\,2\,3)$	9	9

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	7	$(1, 1, (2\,3))$
3	13	$(1, 1, (1\,2\,3))$
6	1	$(1, 1, 1)$

Comparison There are 48 framed tuples and 21 unframed components. The groupoid cardinality is 8. Fixing the first holonomy at $(2\,3)$ leaves 4 framed tuples, whereas fixing it at the identity leaves 18.

17.23 G23. Three commuting returns in D8

Three 2-cells impose pairwise commutation. In a 1-type no further choice of 3-cell is required to classify the matching groupoid.

Presentation Generators are a, b, c, with $aba^{-1}b^{-1} = 1$, $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	176
s	2	4	64
r	2	4	64
rs	2	4	64
r^2	1	8	176

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	40	22
s	16	16
r	16	16
rs	16	16
r^2	40	22

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	84	$(1, 1, s)$
8	8	$(1, 1, 1)$

Comparison There are 176 framed tuples and 92 unframed components. The groupoid cardinality is 22. Fixing the first holonomy at s leaves 16 framed tuples, whereas fixing it at the identity leaves 40.

17.24 G24. Three commuting returns in Q8

Three 2-cells impose pairwise commutation. In a 1-type no further choice of 3-cell is required to classify the matching groupoid.

Presentation Generators are a, b, c, with $aba^{-1}b^{-1} = 1$, $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	176
i	2	4	64
j	2	4	64
k	2	4	64
-1	1	8	176

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	40	22
i	16	16
j	16	16
k	16	16
-1	40	22

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	84	$(1, 1, i)$
8	8	$(1, 1, 1)$

Comparison There are 176 framed tuples and 92 unframed components. The groupoid cardinality is 22. Fixing the first holonomy at i leaves 16 framed tuples, whereas fixing it at the identity leaves 40.

17.25 G25. Three commuting returns in A4

Three 2-cells impose pairwise commutation. In a 1-type no further choice of 3-cell is required to classify the matching groupoid.

Presentation Generators are a, b, c, with $aba^{-1}b^{-1} = 1$, $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	168
(2 3 4)	4	3	27
(2 4 3)	4	3	27
(1 2)(3 4)	3	4	64

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	48	14
(2 3 4)	9	9
(2 4 3)	9	9
(1 2)(3 4)	16	16

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
3	26	$(1, 1, (2\,3\,4))$
4	21	$(1, 1, (1\,2)(3\,4))$
12	1	$(1, 1, 1)$

Comparison There are 168 framed tuples and 48 unframed components. The groupoid cardinality is 14. Fixing the first holonomy at (2 3 4) leaves 9 framed tuples, whereas fixing it at the identity leaves 48.

17.26 G26. Twisted return in C6

Transport by the first loop reverses the second loop's orientation. For an abelian target the condition reduces to a square equal to the identity, but nonabelian transport can enforce reversal by conjugation.

Presentation Generators are a, b, with $aba^{-1}b = 1$. The target is C_6 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	12
r	1	6	12
r^2	1	6	12
r^3	1	6	12
r^4	1	6	12
r^5	1	6	12

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	2	2
r	2	2
r^2	2	2
r^3	2	2
r^4	2	2
r^5	2	2

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
6	12	$(1, 1)$

Comparison There are 12 framed tuples and 12 unframed components. The groupoid cardinality is 2. Fixing the first holonomy at 1 leaves 2 framed tuples, whereas fixing it at the identity leaves 2.

17.27 G27. Twisted return in S3

Transport by the first loop reverses the second loop's orientation. For an abelian

target the condition reduces to a square equal to the identity, but nonabelian transport can enforce reversal by conjugation.

Presentation Generators are a, b, with $aba^{-1}b = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	18
(2 3)	3	2	4
(1 2 3)	2	3	3

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	4	2
(2 3)	4	3
(1 2 3)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	1	$((2\,3), (1\,2\,3))$
2	3	$(1, (2\,3))$
3	1	$((1\,2\,3), 1)$
6	1	$(1, 1)$

Comparison There are 18 framed tuples and 6 unframed components. The groupoid cardinality is 3. Fixing the first holonomy at (2 3) leaves 4 framed tuples, whereas fixing it at the identity leaves 4.

17.28 G28. Twisted return in D8

Transport by the first loop reverses the second loop's orientation. For an abelian target the condition reduces to a square equal to the identity, but nonabelian transport can enforce reversal by conjugation.

Presentation Generators are a, b, with $aba^{-1}b = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	40
s	2	4	16
r	2	4	8
rs	2	4	16
r^2	1	8	40

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	6	4
s	6	5
r	2	2
rs	6	5
r^2	6	4

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	2	(s, r)
4	14	$(1, s)$
8	4	$(1, 1)$

Comparison There are 40 framed tuples and 20 unframed components. The groupoid cardinality is 5. Fixing the first holonomy at s leaves 6 framed tuples, whereas fixing it at the identity leaves 6.

17.29 G29. Twisted return in Q8

Transport by the first loop reverses the second loop's orientation. For an abelian target the condition reduces to a square equal to the identity, but nonabelian transport can enforce reversal by conjugation.

Presentation Generators are a, b, with $aba^{-1}b = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	40
i	2	4	8
j	2	4	8
k	2	4	8
-1	1	8	40

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	2	2
i	6	4
j	6	4
k	6	4
-1	2	2

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	6	(i, j)
4	6	$(i, 1)$
8	4	$(1, 1)$

Comparison There are 40 framed tuples and 16 unframed components. The groupoid cardinality is 5. Fixing the first holonomy at i leaves 6 framed tuples, whereas fixing it at the identity leaves 2.

17.30 G30. Twisted return in D10

Transport by the first loop reverses the second loop's orientation. For an abelian target the condition reduces to a square equal to the identity, but nonabelian transport can enforce reversal by conjugation.

Presentation Generators are a, b, with $aba^{-1}b = 1$. The target is D_{10} of order 10. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	10	40
s	5	2	4
r	2	5	5
r^2	2	5	5

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	6	2
s	6	4
r	1	1
r^2	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	2	(s, r)
2	3	$(1, s)$
5	2	$(r, 1)$
10	1	$(1, 1)$

Comparison There are 40 framed tuples and 8 unframed components. The groupoid cardinality is 4. Fixing the first holonomy at s leaves 6 framed tuples, whereas fixing it at the identity leaves 6.

17.31 G31. Order-two and order-three handles in C6

A separate face constrains the order of each handle. It does not impose any relation between the handles. The conjugation action couples the two coordinate choices.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$. The target is C_6 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	6
r	1	6	6
r^2	1	6	6
r^3	1	6	6
r^4	1	6	6
r^5	1	6	6

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	3	3
r	0	0
r^2	0	0
r^3	3	3
r^4	0	0
r^5	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
6	6	$(1, 1)$

Comparison There are 6 framed tuples and 6 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at 1 leaves 3 framed tuples, whereas fixing it at the identity leaves 3.

17.32 G32. Order-two and order-three handles in S3

A separate face constrains the order of each handle. It does not impose any relation between the handles. The conjugation action couples the two coordinate choices.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	12
(2 3)	3	2	2
(1 2 3)	2	3	3

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	3	2
(2 3)	3	2
(1 2 3)	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	1	$((2\,3),(1\,2\,3))$
2	1	$((2\,3),1)$
3	1	$(1,(1\,2\,3))$
6	1	$(1,1)$

Comparison There are 12 framed tuples and 4 unframed components. The groupoid cardinality is 2. Fixing the first holonomy at (2 3) leaves 3 framed tuples, whereas fixing it at the identity leaves 3.

17.33 G33. Order-two and order-three handles in D8

A separate face constrains the order of each handle. It does not impose any relation between the handles. The conjugation action couples the two coordinate choices.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	6
s	2	4	4
r	2	4	2
rs	2	4	4
r^2	1	8	6

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
s	1	1
r	0	0
rs	1	1
r^2	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	2	$(s, 1)$
8	2	$(1, 1)$

Comparison There are 6 framed tuples and 4 unframed components. The groupoid cardinality is 3/4. Fixing the first holonomy at s leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.34 G34. Order-two and order-three handles in Q8

A separate face constrains the order of each handle. It does not impose any relation between the handles. The conjugation action couples the two coordinate choices.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	2
i	2	4	2
j	2	4	2
k	2	4	2
-1	1	8	2

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
i	0	0
j	0	0
k	0	0
-1	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
8	2	$(1, 1)$

Comparison There are 2 framed tuples and 2 unframed components. The groupoid cardinality is 1/4. Fixing the first holonomy at i leaves 0 framed tuples, whereas fixing it at the identity leaves 1.

17.35 G35. Order-two and order-three handles in A4

A separate face constrains the order of each handle. It does not impose any relation between the handles. The conjugation action couples the two coordinate choices.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	36
(2 3 4)	4	3	3
(2 4 3)	4	3	3
(1 2)(3 4)	3	4	4

Effect of framing one return Fix the label of the first generator at the named repre-

sentative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	9	3
(2 3 4)	0	0
(2 4 3)	0	0
(1 2)(3 4)	9	3

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	2	((1 2)(3 4), (2 3 4))
3	2	(1, (2 3 4))
4	1	((1 2)(3 4), 1)
12	1	(1, 1)

Comparison There are 36 framed tuples and 6 unframed components. The groupoid cardinality is 3. Fixing the first holonomy at (2 3 4) leaves 0 framed tuples, whereas fixing it at the identity leaves 9.

17.36 G36. Triangle closure of type 2, 3, 3 in S3

The first two faces impose order constraints, and the third constrains the transport around their concatenation. The third equation is tested after selecting both labels.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $ababab = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	3
(2 3)	3	2	1
(1 2 3)	2	3	3

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	3	2
(2 3)	0	0
(1 2 3)	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
3	1	(1, (1 2 3))
6	1	(1, 1)

Comparison There are 3 framed tuples and 2 unframed components. The groupoid cardinality is 1/2. Fixing the first holonomy at (2 3) leaves 0 framed tuples, whereas fixing it at the identity leaves 3.

17.37 G37. Triangle closure of type 2, 3, 3 in D8

The first two faces impose order constraints, and the third constrains the transport around their concatenation. The third equation is tested after selecting both labels.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $ababab = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	1
s	2	4	1
r	2	4	1
rs	2	4	1
r^2	1	8	1

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
s	0	0
r	0	0
rs	0	0
r^2	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
8	1	$(1,1)$

Comparison There are 1 framed tuples and 1 unframed components. The groupoid cardinality is $1/8$. Fixing the first holonomy at s leaves 0 framed tuples, whereas fixing it at the identity leaves 1.

17.38 G38. Triangle closure of type 2, 3, 3 in Q8

The first two faces impose order constraints, and the third constrains the transport around their concatenation. The third equation is tested after selecting both labels.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $ababab = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	1
i	2	4	1
j	2	4	1
k	2	4	1
-1	1	8	1

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
i	0	0
j	0	0
k	0	0
-1	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
8	1	$(1,1)$

Comparison There are 1 framed tuples and 1 unframed components. The groupoid cardinality is $1/8$. Fixing the

first holonomy at i leaves 0 framed tuples, whereas fixing it at the identity leaves 1.

17.39 G39. Triangle closure of type 2, 3, 3 in A4

The first two faces impose order constraints, and the third constrains the transport around their concatenation. The third equation is tested after selecting both labels.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $ababab = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	33
(2 3 4)	4	3	3
(2 4 3)	4	3	3
(1 2)(3 4)	3	4	1

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	9	3
(2 3 4)	0	0
(2 4 3)	0	0
(1 2)(3 4)	8	2

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	2	$((1\,2)(3\,4), (2\,3\,4))$
3	2	$(1, (2\,3\,4))$
12	1	$(1, 1)$

Comparison There are 33 framed tuples and 5 unframed components. The groupoid cardinality is $11/4$. Fixing the first holonomy at $(2\,3\,4)$ leaves 0 framed tuples, whereas fixing it at the identity leaves 9.

17.40 G40. Triangle closure of type 2, 3, 2 in C6

The closing face now bounds the square of the concatenation. The local order equations are the same as in the other triangle profile. The gluing count depends on the closing relation.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $abab = 1$. The target is C_6 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	2
r	1	6	2
r^2	1	6	2
r^3	1	6	2
r^4	1	6	2
r^5	1	6	2

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
r	0	0
r^2	0	0
r^3	1	1
r^4	0	0
r^5	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
6	2	$(1, 1)$

Comparison There are 2 framed tuples and 2 unframed components. The groupoid cardinality is 1/3. Fixing the first holonomy at 1 leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.41 G41. Triangle closure of type 2, 3, 2 in S3

The closing face now bounds the square of the concatenation. The local order equations are the same as in the other triangle profile. The gluing count depends on the closing relation.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $abab = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	10
(2 3)	3	2	2
(1 2 3)	2	3	1

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3)	3	2
(1 2 3)	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	1	$((2\,3), (1\,2\,3))$
2	1	$((2\,3), 1)$
6	1	$(1, 1)$

Comparison There are 10 framed tuples and 3 unframed components. The groupoid cardinality is 5/3. Fixing the first holonomy at (2 3) leaves 3 framed tuples, whereas fixing it at the identity leaves 1.

17.42 G42. Triangle closure of type 2, 3, 2 in D10

The closing face now bounds the square of the concatenation. The local order equations are the same as in the other triangle profile. The gluing count depends on the closing relation.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $abab = 1$. The target is D_{10} of order 10. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the

representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	10	6
s	5	2	2
r	2	5	1
r^2	2	5	1

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
s	1	1
r	0	0
r^2	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	1	$(s, 1)$
10	1	$(1, 1)$

Comparison There are 6 framed tuples and 2 unframed components. The groupoid cardinality is 3/5. Fixing the first holonomy at s leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.43 G43. Triangle closure of type 2, 3, 2 in A4

The closing face now bounds the square of the concatenation. The local order equations are the same as in the other triangle profile. The gluing count depends on the closing relation.

Presentation Generators are a, b, with $aa = 1$, $bbb = 1$, $abab = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	4
(2 3 4)	4	3	1
(2 4 3)	4	3	1
(1 2)(3 4)	3	4	4

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3 4)	0	0
(2 4 3)	0	0
(1 2)(3 4)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	1	$((1\,2)(3\,4), 1)$
12	1	$(1, 1)$

Comparison There are 4 framed tuples and 2 unframed components. The groupoid cardinality is 1/3. Fixing the first holonomy at (2 3 4) leaves 0 framed tuples, whereas fixing it at the identity leaves 1.

17.44 G44. Braid comparison in S3

A two-dimensional relation identifies two sequences of three alternating comparisons. The labels can be noncommuting while still satisfying this relation.

Presentation Generators are a, b, with $abab^{-1}a^{-1}b^{-1} = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	12
(2 3)	3	2	2
(1 2 3)	2	3	3

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3)	3	2
(1 2 3)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	1	$((2\,3),(1\,2))$
2	1	$((2\,3),(2\,3))$
3	1	$((1\,2\,3),(1\,2\,3))$
6	1	$(1,1)$

Comparison There are 12 framed tuples and 4 unframed components. The groupoid cardinality is 2. Fixing the first holonomy at (2 3) leaves 3 framed tuples, whereas fixing it at the identity leaves 1.

17.45 G45. Braid comparison in D8

A two-dimensional relation identifies two sequences of three alternating comparisons. The labels can be noncommuting while still satisfying this relation.

Presentation Generators are a, b, with $abab^{-1}a^{-1}b^{-1} = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	8
s	2	4	4
r	2	4	4
rs	2	4	4
r^2	1	8	8

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
s	1	1
r	1	1
rs	1	1
r^2	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	3	(s, s)
8	2	$(1, 1)$

Comparison There are 8 framed tuples and 5 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at s leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.46 G46. Braid comparison in Q8

A two-dimensional relation identifies two sequences of three alternating comparisons. The labels can be noncommuting while still satisfying this relation.

Presentation Generators are a, b, with $abab^{-1}a^{-1}b^{-1} = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	8
i	2	4	4
j	2	4	4
k	2	4	4
-1	1	8	8

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
i	1	1
j	1	1
k	1	1
-1	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	3	(i, i)
8	2	$(1, 1)$

Comparison There are 8 framed tuples and 5 unframed components. The groupoid cardinality is 1. Fixing the first holonomy at i leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.47 G47. Braid comparison in A4

A two-dimensional relation identifies two sequences of three alternating comparisons. The labels can be noncommuting while still satisfying this relation.

Presentation Generators are a, b, with $abab^{-1}a^{-1}b^{-1} = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	36
(2 3 4)	4	3	3
(2 4 3)	4	3	3
(1 2)(3 4)	3	4	4

Effect of framing one return Fix the label of the first generator at the named repre-

sentative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3 4)	4	2
(2 4 3)	4	2
(1 2)(3 4)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	2	$((2\,3\,4),(1\,2\,4))$
3	2	$((2\,3\,4),(2\,3\,4))$
4	1	$((1\,2)(3\,4),(1\,2)(3\,4))$
12	1	$(1,1)$

Comparison There are 36 framed tuples and 6 unframed components. The groupoid cardinality is 3. Fixing the first holonomy at (2 3 4) leaves 4 framed tuples, whereas fixing it at the identity leaves 1.

17.48 G48. Paired commutator closure in S3

Two loops of comparison produce commutators with reciprocal values. Neither pair need commute on its own. The product is required to close at the shared frame.

Presentation Generators are a, b, c, d, with $aba^{-1}b^{-1}cdc^{-1}d^{-1} = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	486
(2 3)	3	2	16
(1 2 3)	2	3	81

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	108	31
(2 3)	72	40
(1 2 3)	81	45

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	60	$(1,(2\,3),1,(1\,2))$
2	15	$(1,1,1,(2\,3))$
3	40	$(1,1,1,(1\,2\,3))$
6	1	$(1,1,1,1)$

Comparison There are 486 framed tuples and 116 unframed components. The groupoid cardinality is 81. Fixing the first holonomy at (2 3) leaves 72 framed tuples, whereas fixing it at the identity leaves 108.

17.49 G49. Paired commutator closure in D8

Two loops of comparison produce commutators with reciprocal values. Neither pair need commute on its own. The product is required to close at the shared frame.

Presentation Generators are a, b, c, d, with $aba^{-1}b^{-1}cdc^{-1}d^{-1} = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	2,176
s	2	4	256
r	2	4	256
rs	2	4	256
r^2	1	8	2,176

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	320	128
s	256	160
r	256	160
rs	256	160
r^2	320	128

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	360	$(1, s, 1, r)$
4	360	$(1, 1, 1, s)$
8	16	$(1, 1, 1, 1)$

Comparison There are 2,176 framed tuples and 736 unframed components. The groupoid cardinality is 272. Fixing the first holonomy at s leaves 256 framed tuples, whereas fixing it at the identity leaves 320.

17.50 G50. Paired commutator closure in Q8

Two loops of comparison produce commutators with reciprocal values. Neither pair need commute on its own. The product is required to close at the shared frame.

Presentation Generators are a, b, c, d, with $aba^{-1}b^{-1}cdc^{-1}d^{-1} = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	2,176
i	2	4	256
j	2	4	256
k	2	4	256
-1	1	8	2,176

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	320	128
i	256	160
j	256	160
k	256	160
-1	320	128

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	360	$(1, i, 1, j)$
4	360	$(1, 1, 1, i)$
8	16	$(1, 1, 1, 1)$

Comparison There are 2,176 framed tuples and 736 unframed components. The groupoid cardinality is 272. Fixing the first holonomy at i leaves 256 framed tuples, whereas fixing it at the identity leaves 320.

17.51 G51. Paired commutator closure in A4

Two loops of comparison produce commutators with reciprocal values. Neither pair need commute on its own. The product is required to close at the shared frame.

Presentation Generators are a, b, c, d, with $aba^{-1}b^{-1}cdc^{-1}d^{-1} = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	5,376
(2 3 4)	4	3	81
(2 4 3)	4	3	81
(1 2)(3 4)	3	4	256

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	576	82
(2 3 4)	432	162
(2 4 3)	432	162
(1 2)(3 4)	448	160

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	400	$(1, (2\,3\,4), 1, (1\,2)(3\,4))$
3	80	$(1, 1, 1, (2\,3\,4))$
4	85	$(1, 1, 1, (1\,2)(3\,4))$
12	1	$(1, 1, 1, 1)$

Comparison There are 5,376 framed tuples and 566 unframed components. The groupoid cardinality is 448. Fixing the first holonomy at (2 3 4) leaves 432 framed tuples, whereas fixing it at the identity leaves 576.

17.52 G52. Central common return in S3

Two independent comparisons each commute with a third, shared reference comparison. There is no 2-cell between the first two loops.

Presentation Generators are a, b, c, with $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	66
(2 3)	3	2	8
(1 2 3)	2	3	27

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	18	8
(2 3)	8	6
(1 2 3)	12	10

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	3	$((2\,3),(1\,2),1)$
2	7	$(1,1,(2\,3))$
3	13	$(1,1,(1\,2\,3))$
6	1	$(1,1,1)$

Comparison There are 66 framed tuples and 24 unframed components. The groupoid cardinality is 11. Fixing the first holonomy at (2 3) leaves 8 framed tuples, whereas fixing it at the identity leaves 18.

17.53 G53. Central common return in D8

Two independent comparisons each commute with a third, shared reference comparison. There is no 2-cell between the first two loops.

Presentation Generators are a, b, c, with $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is D_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	224
s	2	4	64
r	2	4	64
rs	2	4	64
r^2	1	8	224

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	40	22
s	24	20
r	24	20
rs	24	20
r^2	40	22

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	12	$(s,r,1)$
4	84	$(1,1,s)$
8	8	$(1,1,1)$

Comparison There are 224 framed tuples and 104 unframed components. The groupoid cardinality is 28. Fixing the first holonomy at s leaves 24 framed tuples, whereas fixing it at the identity leaves 40.

17.54 G54. Central common return in Q8

Two independent comparisons each commute with a third, shared reference comparison. There is no 2-cell between the first two loops.

Presentation Generators are a, b, c, with $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target

is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	224
i	2	4	64
j	2	4	64
k	2	4	64
-1	1	8	224

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	40	22
i	24	20
j	24	20
k	24	20
-1	40	22

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
2	12	$(i, j, 1)$
4	84	$(1, 1, i)$
8	8	$(1, 1, 1)$

Comparison There are 224 framed tuples and 104 unframed components. The groupoid cardinality is 28. Fixing the first holonomy at i leaves 24 framed tuples, whereas fixing it at the identity leaves 40.

17.55 G55. Central common return in A4

Two independent comparisons each commute with a third, shared reference comparison. There is no 2-cell between the first two loops.

Presentation Generators are a, b, c, with $aca^{-1}c^{-1} = 1$, $bcb^{-1}c^{-1} = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	264
(2 3 4)	4	3	27
(2 4 3)	4	3	27
(1 2)(3 4)	3	4	64

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	48	14
(2 3 4)	18	12
(2 4 3)	18	12
(1 2)(3 4)	24	18

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	8	$((2\,3\,4),(1\,2)(3\,4),1)$
3	26	$(1,1,(2\,3\,4))$
4	21	$(1,1,(1\,2)(3\,4))$
12	1	$(1,1,1)$

Comparison There are 264 framed tuples and 56 unframed components. The groupoid cardinality is 22. Fixing the first holonomy at $(2\,3\,4)$ leaves 18 framed tuples, whereas fixing it at the identity leaves 48.

17.56 G56. Two involutions with a threefold product in C6

The two local labels square to the identity. Their product has order dividing three, which further constrains their relative placement.

Presentation Generators are a, b, with $aa = 1$, $bb = 1$, $ababab = 1$. The target is C_6 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	2
r	1	6	2
r^2	1	6	2
r^3	1	6	2
r^4	1	6	2
r^5	1	6	2

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
r	0	0
r^2	0	0
r^3	1	1
r^4	0	0
r^5	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
6	2	$(1,1)$

Comparison There are 2 framed tuples and 2 unframed components. The groupoid cardinality is 1/3. Fixing the first holonomy at 1 leaves 1 framed tuples, whereas fixing it at the identity leaves 1.

17.57 G57. Two involutions with a threefold product in S3

The two local labels square to the identity. Their product has order dividing three, which further constrains their relative placement.

Presentation Generators are a, b, with $aa = 1$, $bb = 1$, $ababab = 1$. The target is S_3 of order 6. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	6	10
$(2\,3)$	3	2	2
$(1\,2\,3)$	2	3	1

Effect of framing one return Fix the label of the first generator at the named repre-

sentative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3)	3	2
(1 2 3)	0	0

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
1	1	$((2\,3),(1\,2))$
2	1	$((2\,3),(2\,3))$
6	1	$(1,1)$

Comparison There are 10 framed tuples and 3 unframed components. The groupoid cardinality is 5/3. Fixing the first holonomy at (2 3) leaves 3 framed tuples, whereas fixing it at the identity leaves 1.

17.58 G58. Two involutions with a threefold product in Q8

The two local labels square to the identity. Their product has order dividing three, which further constrains their relative placement.

Presentation Generators are a, b, with $aa = 1$, $bb = 1$, $ababab = 1$. The target is Q_8 of order 8. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	8	2
i	2	4	2
j	2	4	2
k	2	4	2
-1	1	8	2

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
i	0	0
j	0	0
k	0	0
-1	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
8	2	$(1,1)$

Comparison There are 2 framed tuples and 2 unframed components. The groupoid cardinality is 1/4. Fixing the first holonomy at i leaves 0 framed tuples, whereas fixing it at the identity leaves 1.

17.59 G59. Two involutions with a threefold product in A4

The two local labels square to the identity. Their product has order dividing three, which further constrains their relative placement.

Presentation Generators are a, b, with $aa = 1$, $bb = 1$, $ababab = 1$. The target is A_4 of order 12. All locally permitted comparisons are group elements.

Burnside check The second column is the conjugacy-class size. The third is the number of target elements commuting with the representative. The last counts tuples fixed under its simultaneous conjugation.

Gauge	Class	Centralizer	Fixed tuples
1	1	12	4
(2 3 4)	4	3	1
(2 4 3)	4	3	1
(1 2)(3 4)	3	4	4

Effect of framing one return Fix the label of the first generator at the named representative in a framed record. The middle column counts completions there. The last column instead counts unframed components where the first label lies anywhere in that class.

Label	Framed	Unframed components
1	1	1
(2 3 4)	0	0
(2 4 3)	0	0
(1 2)(3 4)	1	1

Automorphism profile A representative tuple is shown for each stabilizer size. The stabilizer is the common centralizer of all its entries.

Order	Components	One tuple
4	1	$((1\,2)(3\,4), (1\,2)(3\,4))$
12	1	$(1, 1)$

Comparison There are 4 framed tuples and 2 unframed components. The groupoid cardinality is 1/3. Fixing the first holonomy at (2 3 4) leaves 0 framed tuples, whereas fixing it at the identity leaves 1.

18 Atlas of closed and open relay diagrams

Matrices list numbers of local records by input row and output column. The relay sequence can be read as open, or closed by requiring the last value to equal the first. The closing trace counts full witnesses. A label count and a witness count differ when a matrix entry is greater than one. See Section 12.

18.1 R01. 2-label rotation with net shift 0

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 1. Relay i has matrix M_i. The ordered matrix product counts 2 open records over all pairs of exposed boundary values.

$$M_1 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right], \quad M_2 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$$

$$M_3 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$$

$$M_1 \cdots M_3 = \left[\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right]$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 2 records. Ignoring internal multiplicities leaves 2 possible cyclic label sequences.

Port	0	1
1	1	1
2	1	1
3	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1
0	0	1
1	1	0

One restriction and repeated circuits Disallowing the pair $0, 1$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	2
2	2
3	2
4	2

18.2 R02. 2-label rotation with net shift 1

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 1. Relay i has matrix M_i. The ordered matrix product counts 2 open records over all pairs of exposed boundary values.

$$M_1 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right], \quad M_2 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$$

$$M_3 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$$

$$M_1 \cdots M_3 = \left[\begin{smallmatrix} 0 & 1 \\ 1 & 0 \end{smallmatrix}\right]$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal multiplicities leaves 0 possible cyclic label sequences.

Port	0	1
1	0	0
2	0	0
3	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1
0	0	0
1	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	2
3	0
4	2

18.3 R03. 3-label rotation with net shift 0

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 3 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	1	1
2	1	1	1
3	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	1	0
1	0	0	1
2	1	0	0

One restriction and repeated circuits Disallowing the pair $0, 1$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	3
3	3
4	3

18.4 R04. 3-label rotation with net shift 1

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal multiplicities leaves 0 possible cyclic label sequences.

Port	0	1	2
1	0	0	0
2	0	0	0
3	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below

each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	0
3	3
4	0

18.5 R05. 3-label rotation with net shift 2

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal multiplicities leaves 0 possible cyclic label sequences.

Port	0	1	2
1	0	0	0
2	0	0	0
3	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	0
3	3
4	0

18.6 R06. 4-label rotation with net shift 0

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2	3
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	1	0	0
1	0	0	1	0
2	0	0	0	1
3	1	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 1$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	4
3	4
4	4

18.7 R07. 4-label rotation with net shift 1

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal multiplicities leaves 0 possible cyclic label sequences.

Port	0	1	2	3
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	0
3	0
4	4

18.8 R08. 4-label rotation with net shift 2

Each relay implements a deterministic rotation of the labels. The closed system requires a fixed point of their composite. Every label has a unique continuation at every open port.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0&0&0&1\\1&0&0&0\\0&1&0&0\\0&0&1&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0&0&1&0\\0&0&0&1\\1&0&0&0\\0&1&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal multiplicities leaves 0 possible cyclic label sequences.

Port	0	1	2	3
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	4
3	0
4	4

18.9 R09. 3-label order of two transpositions

Two local swaps alternate, returning to the same physical port. The composite need not have order two.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0&1&0\\1&0&0\\0&0&1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1&0&0\\0&0&1\\0&1&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1&0\\1&0&0\\0&0&1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1&0&0\\0&0&1\\0&1&0 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 0&1&0\\0&0&1\\1&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal

multiplicities leaves 0 possible cyclic label sequences.

Port	0	1	2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	0
3	3
4	0

18.10 R10. 3-label conjugating a shift

The two shifts would cancel in isolation. The intervening swap changes their action on the label seen after one circuit.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 0 records. Ignoring internal multiplicities leaves 0 possible cyclic label sequences.

Port	0	1	2
1	0	0	0
2	0	0	0
3	0	0	0
4	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	0	0	0
2	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 0 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	0
2	0
3	3
4	0

18.11 R11. 3-label one optional swap

A relay can preserve its label or swap two values. A fixed label can have more than one internal witness for an unchanged port value.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 6 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	1	2
2	1	1	2
3	1	2	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	1	0
1	1	0	0
2	0	0	2

One restriction and repeated circuits Disallowing the pair $2, 2$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	8
3	16
4	32

18.12 R12. 3-label two optional swaps

Two local choices are composed before testing a return. A choice that is possible on an isolated link may have no global completion on the cycle.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 12 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 2 & 1 \\ 1 & 2 & 1 \\ 2 & 0 & 2 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 5 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	2	2
2	2	1	2
3	2	2	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	1	0
1	2	0	0
2	0	0	2

One restriction and repeated circuits Disallowing the pair $1, 0$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	5
2	17
3	65
4	257

18.13 R13. 4-label order of two transpositions

Two local swaps alternate, returning to the same physical port. The composite need not have order two.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 1&0&0&0 \\ 0&0&0&1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2	3
1	0	0	0	1
2	0	0	0	1
3	0	0	0	1
4	0	0	0	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	1

One restriction and repeated circuits Disallowing the pair $3, 3$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	1
3	4
4	1

18.14 R14. 4-label conjugating a shift

The two shifts would cancel in isolation. The intervening swap changes their action on the label seen after one circuit.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1&0&0 \\ 1&0&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 0&0&0&1 \\ 1&0&0&0 \\ 0&1&0&0 \\ 0&0&1&0 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 0&1&0&0 \\ 0&0&0&1 \\ 0&0&1&0 \\ 1&0&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2	3
1	0	0	1	0
2	0	0	1	0
3	0	0	0	1
4	0	0	0	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	0	0	1	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $2, 2$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	1
3	4
4	1

18.15 R15. 4-label one optional swap

A relay can preserve its label or swap two values. A fixed label can have more than one internal witness for an unchanged port value.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 8 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1&1&0&0 \\ 1&1&0&0 \\ 0&0&2&0 \\ 0&0&0&2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1&0&0&0 \\ 0&0&1&0 \\ 0&1&0&0 \\ 0&0&0&1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1&0&0 \\ 0&0&1&0 \\ 0&0&0&1 \\ 1&0&0&0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0&1&0&1 \\ 0&1&0&1 \\ 0&0&2&0 \\ 2&0&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 3 records. Ignoring internal multiplicities leaves 2 possible cyclic label sequences.

Port	0	1	2	3
1	0	1	2	0
2	1	0	2	0
3	1	2	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	0	0	2	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $2, 2$ at relay 1 removes 2 closed records. In the table below

each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	9
3	15
4	33

18.16 R16. 4-label two optional swaps

Two local choices are composed before testing a return. A choice that is possible on an isolated link may have no global completion on the cycle.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 16 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 2 & 1 & 1 \\ 0 & 2 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 4 & 0 & 0 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 2 possible cyclic label sequences.

Port	0	1	2	3
1	0	2	2	0
2	2	0	2	0
3	2	2	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	2	0	0	0
2	0	0	2	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $1, 0$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	16
3	64
4	256

18.17 R17. 3-state monotone loop

Every step retains or increases the label by one. Closing the loop forces equality at each step, because strict increases cannot be undone.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 12 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 3 & 3 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated

port in the closed network. Its counts by value sum to 3 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	1	1
2	1	1	1
3	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	1	0	0
1	0	1	0
2	0	0	1

One restriction and repeated circuits Disallowing the pair $0,0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	3
3	3
4	3

18.18 R18. 3-state one reverse window

The first two relays never decrease their labels, and the last never increases them. Closing the loop can now witness one outward step and its return.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 15 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 3 & 3 & 1 \\ 1 & 3 & 2 \\ 0 & 1 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 7 records. Ignoring internal multiplicities leaves 7 possible cyclic label sequences.

Port	0	1	2
1	3	3	1
2	2	3	2
3	1	3	3

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	2	1	0
1	0	2	1
2	0	0	1

One restriction and repeated circuits Disallowing the pair $0,0$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	7
2	29
3	136
4	669

18.19 R19. 3-state two reverse windows

The monotonicity direction alternates. A loop can return through either reverse window, so a port value does not determine the intermediate path.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 29 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 5 & 4 & 1 \\ 4 & 6 & 3 \\ 1 & 3 & 2 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 13 records. Ignoring internal multiplicities leaves 13 possible cyclic label sequences.

Port	0	1	2
1	5	6	2
2	2	6	5
3	5	6	2
4	2	6	5

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	2	3	0
1	0	3	3
2	0	0	2

One restriction and repeated circuits Disallowing the pair $0, 1$ at relay 1 removes 3 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	13
2	117
3	1,186
4	12,389

18.20 R20. 3-state restricted middle state

One relay allows only the bottom and top states. It exposes a filter at one port rather than altering the allowed successor relation of the other relays.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 5 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 3 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	1	1
2	1	0	2
3	1	0	2

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	1	0	0
1	0	0	1
2	0	0	1

One restriction and repeated circuits Disallowing the pair $0,0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	5
3	9
4	17

18.21 R21. 3-state weighted stable observations

Different states carry different numbers of evidence records. Every internal certificate still has the same prescribed input and output labels.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 21 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 3 & 2 & 0 \\ 2 & 5 & 3 \\ 0 & 3 & 3 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 11 records. Ignoring internal multiplicities leaves 5 possible cyclic label sequences.

Port	0	1	2
1	3	5	3
2	1	4	6
3	1	4	6

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	1	2	0
1	0	2	3
2	0	0	3

One restriction and repeated circuits Disallowing the pair $1,2$ at relay 1 removes 3 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	11
2	69
3	491
4	3,673

18.22 R22. 4-state monotone loop

Every step retains or increases the label by one. Closing the loop forces equality at each step, because strict increases cannot be undone.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 20 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 3 & 3 & 1 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & 1 & 3 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2	3
1	1	1	1	1
2	1	1	1	1
3	1	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	1	0	0	0
1	0	1	0	0
2	0	0	1	0
3	0	0	0	1

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	4
3	4
4	4

18.23 R23. 4-state one reverse window

The first two relays never decrease their labels, and the last never increases them. Closing the loop can now witness one outward step and its return.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 23 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 3 & 3 & 1 & 0 \\ 1 & 3 & 3 & 1 \\ 0 & 1 & 3 & 2 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 10 records. Ignoring internal multiplicities leaves 10 possible cyclic label sequences.

Port	0	1	2	3
1	3	3	3	1
2	2	3	3	2
3	1	3	3	3

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	2	1	0	0
1	0	2	1	0
2	0	0	2	1
3	0	0	0	1

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	10
2	44
3	220
4	1,164

18.24 R24. 4-state two reverse windows

The monotonicity direction alternates. A loop can return through either reverse window, so a port value does not determine the intermediate path.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 45 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 5 & 4 & 1 & 0 \\ 4 & 6 & 4 & 1 \\ 1 & 4 & 6 & 3 \\ 0 & 1 & 3 & 2 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 19 records. Ignoring internal multiplicities leaves 19 possible cyclic label sequences.

Port	0	1	2	3
1	5	6	6	2
2	2	6	6	5
3	5	6	6	2
4	2	6	6	5

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	2	3	0	0
1	0	3	3	0
2	0	0	3	3
3	0	0	0	2

One restriction and repeated circuits Disallowing the pair $0, 1$ at relay 1 removes 3 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	19
2	187
3	2,110
4	25,147

18.25 R25. 4-state restricted middle state

One relay allows only the bottom and top states. It exposes a filter at one port rather than altering the allowed successor relation of the other relays.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 5 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 3 records. Ignoring internal

multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2	3
1	1	0	1	1
2	1	0	0	2
3	1	0	0	2

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	1
3	0	0	0	1

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	5
3	9
4	17

18.26 R26. 4-state weighted stable observations

Different states carry different numbers of evidence records. Every internal certificate still has the same prescribed input and output labels.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 37 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 3 & 2 & 0 & 0 \\ 2 & 5 & 3 & 0 \\ 0 & 3 & 7 & 4 \\ 0 & 0 & 4 & 4 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 19 records. Ignoring internal multiplicities leaves 7 possible cyclic label sequences.

Port	0	1	2	3
1	3	5	7	4
2	1	4	6	8
3	1	4	6	8

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	1	2	0	0
1	0	2	3	0
2	0	0	3	4
3	0	0	0	4

One restriction and repeated circuits Disallowing the pair $2, 3$ at relay 1 removes 4 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	19
2	157
3	1,507
4	15,449

18.27 R27. 3-label reset and return

A reset forgets its previous label. The next rotation makes the label at the opening port observable again.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2
1	0	1	0
2	0	0	1
3	1	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	0	0	1
2	0	0	0

One restriction and repeated circuits Disallowing the pair $1, 2$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	1
3	1
4	1

18.28 R28. 3-label split then reset

The first relay can start only at state zero but can choose any output. A later reset closes several different internal records at the same opening port.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 3 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	3	0	0
2	1	1	1
3	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	1	1	1
1	0	0	0
2	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	9
3	27
4	81

18.29 R29. 3-label reset then split

The early reset puts the next relay in its uniquely enabled state. Reordering these two operations changes the allowed intermediate label, even if the total counts coincide.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 9 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 3 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	1	1
2	3	0	0
3	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	1	0	0
1	1	0	0
2	1	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	3
2	9
3	27
4	81

18.30 R30. 3-label parallel paths across a reset

Two relays each permit a stable or a shifted record. The reset discards one interface coordinate but not the number of distinct paths through it.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 12 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 2 & 2 & 0 \\ 2 & 2 & 0 \\ 2 & 2 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2
1	2	2	0
2	1	2	1
3	4	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	1	1	0
1	0	1	1
2	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	16
3	64
4	256

18.31 R31. 3-label one recovering alternate

The first stage may follow the normal rotation or reset to a reference label. Closing the loop exposes which normal and recovery witnesses are feasible.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 6 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 2 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2
1	1	1	2
2	2	1	1
3	1	2	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	1	0
1	0	0	1
2	2	0	0

One restriction and repeated circuits Disallowing the pair $2, 0$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	6
3	10
4	18

18.32 R32. 3-label checkpoint filter after reset

The final checkpoint admits only its first two labels. The boundary can be feasible because the reset sends all inputs to a suitable next state.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 3 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2
1	0	1	0
2	1	0	0
3	0	1	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	1	0	0
2	0	0	0

One restriction and repeated circuits Disallowing the pair $1, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	1
3	1
4	1

18.33 R33. 4-label reset and return

A reset forgets its previous label. The next rotation makes the label at the opening port observable again.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2	3
1	0	1	0	0
2	0	0	1	0
3	1	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	0	1	0
2	0	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $1, 2$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	1
3	1
4	1

18.34 R34. 4-label split then reset

The first relay can start only at state zero but can choose any output. A later reset closes several different internal records at the same opening port.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 4 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2	3
1	4	0	0	0
2	1	1	1	1
3	1	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	1	1	1	1
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	16
3	64
4	256

18.35 R35. 4-label reset then split

The early reset puts the next relay in its uniquely enabled state. Reordering these two operations changes the allowed intermediate label, even if the total counts coincide.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 16 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1&0&0&0\\ 1&0&0&0\\ 1&0&0&0\\ 1&0&0&0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1&1&1&1\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1&0&0\\ 0&0&1&0\\ 0&0&0&1\\ 1&0&0&0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1\\ 1&1&1&1 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2	3
1	1	1	1	1
2	4	0	0	0
3	1	1	1	1

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	1	0	0	0
1	1	0	0	0
2	1	0	0	0
3	1	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	16
3	64
4	256

18.36 R36. 4-label parallel paths across a reset

Two relays each permit a stable or a shifted record. The reset discards one interface coordinate but not the number of distinct paths through it.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 16 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1&1&0&0\\ 0&1&1&0\\ 0&0&1&1\\ 1&0&0&1 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1&0&0&0\\ 1&0&0&0\\ 1&0&0&0\\ 1&0&0&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1&1&0&0\\ 0&1&1&0\\ 0&0&1&1\\ 1&0&0&1 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 2&2&0&0\\ 2&2&0&0\\ 2&2&0&0\\ 2&2&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 4 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2	3
1	2	2	0	0
2	1	2	1	0
3	4	0	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	1	1	0	0
1	0	1	1	0
2	0	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $0, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	4
2	16
3	64
4	256

18.37 R37. 4-label one recovering alternate

The first stage may follow the normal rotation or reset to a reference label. Closing the loop exposes which normal and recovery witnesses are feasible.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 8 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 2 & 0 & 0 & 0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2	3
1	0	0	1	0
2	1	0	0	0
3	0	1	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	0	0	0
2	1	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $2, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	3
3	7
4	19

18.38 R38. 4-label checkpoint filter after reset

The final checkpoint admits only its first two labels. The boundary can be feasible because the reset sends all inputs to a suitable next state.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 4 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 1&0&0&0\\1&0&0&0\\1&0&0&0\\1&0&0&0 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 0&1&0&0\\0&0&1&0\\0&0&0&1\\1&0&0&0 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 1&0&0&0\\0&1&0&0\\0&0&0&0\\0&0&0&0 \end{bmatrix}$$

$$M_1 \cdots M_3 = \begin{bmatrix} 0&1&0&0\\0&1&0&0\\0&1&0&0\\0&1&0&0 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 1 records. Ignoring internal multiplicities leaves 1 possible cyclic label sequences.

Port	0	1	2	3
1	0	1	0	0
2	1	0	0	0
3	0	1	0	0

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	1	0	0	0
2	0	0	0	0
3	0	0	0	0

One restriction and repeated circuits Disallowing the pair $1, 0$ at relay 1 removes 1 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	1
2	1
3	1
4	1

18.39 R39. 2-state merged evidence branches

Multiple local branches can produce the same port pair. Those branches remain separate witnesses during composition. Replacing positive entries by one tests how much the count depends on internal evidence multiplicity.

Open wiring The labels are 0 through 1. Relay i has matrix M_i. The ordered matrix product counts 16 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 3&2\\1&2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1&0\\1&1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0&1\\1&0 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1&1\\0&1 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 2&7\\2&5 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 7 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1
1	2	5
2	1	6
3	3	4
4	4	3

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1
0	0	2
1	1	4

One restriction and repeated circuits Disallowing the pair $1, 1$ at relay 1 removes 4 closed records. In the table below

each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	7
2	57
3	427
4	3,217

18.40 R40. 3-state merged evidence branches

Multiple local branches can produce the same port pair. Those branches remain separate witnesses during composition. Replacing positive entries by one tests how much the count depends on internal evidence multiplicity.

Open wiring The labels are 0 through 2. Relay i has matrix M_i. The ordered matrix product counts 32 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 3 & 2 & 0 \\ 0 & 2 & 2 \\ 1 & 0 & 2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 0 & 5 & 7 \\ 2 & 4 & 6 \\ 2 & 3 & 3 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 7 records. Ignoring internal multiplicities leaves 4 possible cyclic label sequences.

Port	0	1	2
1	0	4	3
2	1	2	4
3	3	2	2
4	2	3	2

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2
0	0	0	0
1	0	2	2
2	1	0	2

One restriction and repeated circuits Disallowing the pair $1,1$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	7
2	109
3	1,021
4	10,529

18.41 R41. 4-state merged evidence branches

Multiple local branches can produce the same port pair. Those branches remain separate witnesses during composition. Replacing positive entries by one tests how much the count depends on internal evidence multiplicity.

Open wiring The labels are 0 through 3. Relay i has matrix M_i. The ordered matrix product counts 48 open records over all pairs of exposed boundary values.

$$M_1 = \begin{bmatrix} 3 & 2 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 2 & 2 \\ 1 & 0 & 0 & 2 \end{bmatrix}, \quad M_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

$$M_3 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, \quad M_4 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M_1 \cdots M_4 = \begin{bmatrix} 0 & 5 & 7 & 2 \\ 0 & 2 & 6 & 6 \\ 2 & 2 & 2 & 6 \\ 2 & 3 & 1 & 2 \end{bmatrix}$$

Pinning a label in feedback Port 1 is the opening port. Each row fixes the indicated port in the closed network. Its counts by value sum to 6 records. Ignoring internal multiplicities leaves 3 possible cyclic label sequences.

Port	0	1	2	3
1	0	2	2	2
2	0	2	2	2
3	2	2	2	0
4	0	2	2	2

First-edge completion table Fix both ports of relay 1. Entries multiply the local multiplicity by the number of ways to return through all remaining relays.

Input	0	1	2	3
0	0	0	0	0
1	0	2	0	0
2	0	0	2	0
3	0	0	0	2

One restriction and repeated circuits Disallowing the pair $1, 1$ at relay 1 removes 2 closed records. In the table below each circuit is an independent copy, with the end of one wired to the start of the next. Only the last copy must return to the initial label.

Circuits	Completions
1	6
2	120
3	1,548
4	15,544

19 Atlas of distributed knowledge

Rows are possible joint records, and each view column gives the class of that record for one observer. Records in the same class are individually indistinguishable. The public signal is announced to all observers together. After seeing a positive signal the universe is restricted to the marked rows. The fixed-point calculation is Theorem 8.2.

19.1 K01. 4-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	+	2	∞
D	2	1	+	+	3	∞
E	2	2	+	+	4	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	4	3
2	3	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	3
1	3	3
2	2	3
3	1	3
4	0	3
5	0	3

Counterexample path The actual record is E. Before the signal the chain below reaches a record where the proposition is false. Its length 4 is the first nesting depth that fails there.

Step	From	Shared view	To
1	E	1	D
2	D	2	C
3	C	1	B
4	B	2	A

Public signal The positive signal admits $\{C, D, E\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.2 K02. 5-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	+	3	∞
E	2	2	+	+	4	∞
F	3	2	+	+	5	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	5	3
2	4	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	5	3
1	4	3
2	3	3
3	2	3
4	1	3
5	0	3
6	0	3

Counterexample path The actual record is F. Before the signal the chain below reaches a record where the proposition is false. Its length 5 is the first nesting depth that fails there.

Step	From	Shared view	To
1	F	2	E
2	E	1	D
3	D	2	C
4	C	1	B
5	B	2	A

Public signal The positive signal admits $\{D, E, F\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.3 K03. 6-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	+	3	∞
E	2	2	+	+	4	∞
F	3	2	+	+	5	∞
G	3	3	+	+	6	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	6	4
2	5	4

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	6	4
1	5	4
2	4	4
3	3	4
4	2	4
5	1	4
6	0	4
7	0	4

Counterexample path The actual record is G. Before the signal the chain below reaches a record where the proposition is false. Its length 6 is the first nesting depth that fails there.

Step	From	Shared view	To
1	G	1	F
2	F	2	E
3	E	1	D
4	D	2	C
5	C	1	B
6	B	2	A

Public signal The positive signal admits $\{D, E, F, G\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.4 K04. 7-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	−	3	−
E	2	2	+	+	4	∞
F	3	2	+	+	5	∞
G	3	3	+	+	6	∞
H	4	3	+	+	7	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	7	4
2	6	4

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	7	4
1	6	4
2	5	4
3	4	4
4	3	4
5	2	4
6	1	4
7	0	4
8	0	4

Counterexample path The actual record is H. Before the signal the chain below reaches a record where the proposition is false. Its length 7 is the first nesting depth that fails there.

Step	From	Shared view	To
1	H	2	G
2	G	1	F
3	F	2	E
4	E	1	D
5	D	2	C
6	C	1	B
7	B	2	A

Public signal The positive signal admits $\{E, F, G, H\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.5 K05. 8-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	−	3	−
E	2	2	+	+	4	∞
F	3	2	+	+	5	∞
G	3	3	+	+	6	∞
H	4	3	+	+	7	∞
I	4	4	+	+	8	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	8	5
2	7	5

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	8	5
1	7	5
2	6	5
3	5	5
4	4	5
5	3	5
6	2	5
7	1	5
8	0	5
9	0	5

Counterexample path The actual record is I. Before the signal the chain below reaches a record where the proposition is false. Its length 8 is the first nesting depth that fails there.

Step	From	Shared view	To
1	I	1	H
2	H	2	G
3	G	1	F
4	F	2	E
5	E	1	D
6	D	2	C
7	C	1	B
8	B	2	A

Public signal The positive signal admits $\{E, F, G, H, I\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.6 K06. 9-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	–	–	0	–
B	1	0	+	–	1	–
C	1	1	+	–	2	–
D	2	1	+	–	3	–
E	2	2	+	–	4	–
F	3	2	+	+	5	∞
G	3	3	+	+	6	∞
H	4	3	+	+	7	∞
I	4	4	+	+	8	∞
J	5	4	+	+	9	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	9	5
2	8	5

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	9	5
1	8	5
2	7	5
3	6	5
4	5	5
5	4	5
6	3	5
7	2	5
8	1	5
9	0	5
10	0	5

Counterexample path The actual record is J. Before the signal the chain below reaches a record where the proposition is false. Its length 9 is the first nesting depth that fails there.

Step	From	Shared view	To
1	J	2	I
2	I	1	H
3	H	2	G
4	G	1	F
5	F	2	E
6	E	1	D
7	D	2	C
8	C	1	B
9	B	2	A

Public signal The positive signal admits $\{F, G, H, I, J\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.7 K07. 10-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	−	3	−
E	2	2	+	−	4	−
F	3	2	+	+	5	∞
G	3	3	+	+	6	∞
H	4	3	+	+	7	∞
I	4	4	+	+	8	∞
J	5	4	+	+	9	∞
K	5	5	+	+	10	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	10	6
2	9	6

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	10	6
1	9	6
2	8	6
3	7	6
4	6	6
5	5	6
6	4	6
7	3	6
8	2	6
9	1	6
10	0	6
11	0	6

Counterexample path The actual record is K. Before the signal the chain below

reaches a record where the proposition is false. Its length 10 is the first nesting depth that fails there.

Step	From	Shared view	To
1	K	1	J
2	J	2	I
3	I	1	H
4	H	2	G
5	G	1	F
6	F	2	E
7	E	1	D
8	D	2	C
9	C	1	B
10	B	2	A

Public signal The positive signal admits $\{F, G, H, I, J, K\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.8 K08. 11-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	−	3	−
E	2	2	+	−	4	−
F	3	2	+	−	5	−
G	3	3	+	+	6	∞
H	4	3	+	+	7	∞
I	4	4	+	+	8	∞
J	5	4	+	+	9	∞
K	5	5	+	+	10	∞
L	6	5	+	+	11	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	11	6
2	10	6

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	11	6
1	10	6
2	9	6
3	8	6
4	7	6
5	6	6
6	5	6
7	4	6
8	3	6
9	2	6
10	1	6
11	0	6
12	0	6

Counterexample path The actual record is L. Before the signal the chain below reaches a record where the proposition is

false. Its length 11 is the first nesting depth that fails there.

Step	From	Shared view	To
1	L	2	K
2	K	1	J
3	J	2	I
4	I	1	H
5	H	2	G
6	G	1	F
7	F	2	E
8	E	1	D
9	D	2	C
10	C	1	B
11	B	2	A

Public signal The positive signal admits $\{G, H, I, J, K, L\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.9 K09. 12-delivery ladder

The records in order count delivered acknowledgments, beginning with no initial delivery. Each acknowledgment is invisible to its sender until another one comes back. The extra public signal certifies that the run passed the indicated lower half of the ladder.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^{+}
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	−	2	−
D	2	1	+	−	3	−
E	2	2	+	−	4	−
F	3	2	+	−	5	−
G	3	3	+	+	6	∞
H	4	3	+	+	7	∞
I	4	4	+	+	8	∞
J	5	4	+	+	9	∞
K	5	5	+	+	10	∞
L	6	5	+	+	11	∞
M	6	6	+	+	12	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	12	7
2	11	7

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	12	7
1	11	7
2	10	7
3	9	7
4	8	7
5	7	7
6	6	7
7	5	7
8	4	7
9	3	7
10	2	7
11	1	7
12	0	7
13	0	7

Counterexample path The actual record is M. Before the signal the chain below

reaches a record where the proposition is false. Its length 12 is the first nesting depth that fails there.

Step	From	Shared view	To
1	M	1	L
2	L	2	K
3	K	1	J
4	J	2	I
5	I	1	H
6	H	2	G
7	G	1	F
8	F	2	E
9	E	1	D
10	D	2	C
11	C	1	B
12	B	2	A

Public signal The positive signal admits $\{G, H, I, J, K, L, M\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.10 K10. Independent status fields

A reports the writer status, while B reports the reader status. The two statuses have no dependency.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	0	1	−	+	0	0
C	0	2	−	+	0	0
D	1	0	−	−	0	−
E	1	1	−	+	0	0
F	1	2	−	+	0	0
G	2	0	+	−	1	−
H	2	1	+	+	1	1
I	2	2	+	+	1	1

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	3	2
2	0	0

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	3	2
1	0	0
2	0	0

Counterexample path The actual record is I. Before the signal the chain below reaches a record where the proposition is false. Its length 1 is the first nesting depth that fails there.

Step	From	Shared view	To
1	I	2	C

Public signal The positive signal admits $\{B, C, E, F, H, I\}$. A counterexample still occurs at distance 1 after restricting to those records.

19.11 K11. Reader never leads writer

The reader can lag but never report a version ahead of the writer. The proposition is that the writer advanced.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	+	2	∞
D	2	0	+	−	1	−
E	2	1	+	+	2	∞
F	2	2	+	+	2	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	5	3
2	3	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	5	3
1	3	3
2	0	3
3	0	3

Counterexample path The actual record is F. Before the signal the chain below reaches a record where the proposition is false. Its length 2 is the first nesting depth that fails there.

Step	From	Shared view	To
1	F	1	D
2	D	2	A

Public signal The positive signal admits $\{C, E, F\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.12 K12. Reader at most one step behind

A bounded lag removes some indistinguishable runs but leaves others connected by alternating views.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	1	0	+	−	1	−
C	1	1	+	+	2	∞
D	2	1	+	+	3	∞
E	2	2	+	+	4	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	4	3
2	3	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	3
1	3	3
2	2	3
3	1	3
4	0	3
5	0	3

Counterexample path The actual record is E. Before the signal the chain below reaches a record where the proposition is false. Its length 4 is the first nesting depth that fails there.

Step	From	Shared view	To
1	E	1	D
2	D	2	C
3	C	1	B
4	B	2	A

Public signal The positive signal admits $\{C, D, E\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.13 K13. Agreement except during handoff

A shared completed initialization separates idle status from handoff status before either observer sees a finish.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	+	0	0
B	1	1	+	+	∞	∞
C	1	2	+	−	∞	−
D	2	1	+	+	∞	∞
E	2	2	+	−	∞	−

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	4	2
2	4	2

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	2
1	4	2

Knowledge certificate Record D has no counterexample in its indistinguishability component even before the announcement.

Public signal The positive signal admits $\{A, B, D\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.14 K14. Cyclic observation relation

Two adjacent local labels can be related by transport on a three-cycle. The only bad record has both labels zero.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	0	1	+	−	1	−
C	1	1	+	+	2	∞
D	1	2	+	+	3	∞
E	2	0	+	+	1	∞
F	2	2	+	+	2	∞

Individual knowledge The count before

the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	4	4
2	4	4

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	5	4
1	3	4
2	1	4
3	0	4
4	0	4

Counterexample path The actual record is D. Before the signal the chain below reaches a record where the proposition is false. Its length 3 is the first nesting depth that fails there.

Step	From	Shared view	To
1	D	1	C
2	C	2	B
3	B	1	A

Public signal The positive signal admits $\{C, D, E, F\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.15 K15. Disjoint mode switch

The final mode is visible to both observers. In the earlier mode the proposition depends on one party's private label.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	0	1	+	+	1	∞
C	1	0	−	−	0	−
D	1	1	+	+	1	∞
E	2	2	+	+	∞	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	1	3
2	3	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	3	3
1	1	3
2	1	3

Counterexample path The actual record is D. Before the signal the chain below reaches a record where the proposition is false. Its length 1 is the first nesting depth that fails there.

Step	From	Shared view	To
1	D	1	C

Public signal The positive signal admits $\{B, D, E\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.16 K16. One faulty observation eliminated

The middle joint observation is disallowed by the environment. The proposition is a threshold of the two remaining coordinates.

Possible runs and observation classes Distance is the shortest indistinguisha-

bility path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	0	1	−	−	0	−
C	0	2	+	−	1	−
D	1	0	−	+	0	0
E	1	2	+	+	1	1
F	2	0	+	+	1	1
G	2	1	+	+	1	2
H	2	2	+	+	2	2

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	3	3
2	3	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	5	4
1	1	2
2	0	0
3	0	0

Counterexample path The actual record is H. Before the signal the chain below reaches a record where the proposition is false. Its length 2 is the first nesting depth that fails there.

Step	From	Shared view	To
1	H	2	C
2	C	1	A

Public signal The positive signal admits $\{D, E, F, G, H\}$. A counterexample still occurs at distance 2 after restricting to those records.

19.17 K17. Lower edge calibrated

Alternating equalities of one coordinate connect the record set into a narrow chain. A public marker can distinguish the first record.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	0	1	+	+	1	∞
C	1	1	+	+	2	∞
D	1	2	+	+	3	∞
E	2	2	+	+	4	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	3	4
2	4	4

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	4
1	3	4
2	2	4
3	1	4
4	0	4
5	0	4

Counterexample path The actual record is E. Before the signal the chain below reaches a record where the proposition is

false. Its length 4 is the first nesting depth that fails there.

Step	From	Shared view	To
1	E	2	D
2	D	1	C
3	C	2	B
4	B	1	A

Public signal The positive signal admits $\{B, C, D, E\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.18 K18. Two matched epochs

One epoch can be observed as an old label at one endpoint. The extra marker rules out a low aggregate observation without disclosing both labels.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	P	Signal	d	d^+
A	0	0	−	−	0	−
B	0	1	−	−	0	−
C	1	0	+	−	1	−
D	1	1	+	+	1	∞
E	2	2	+	+	2	∞
F	2	1	+	+	1	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	4	3
2	1	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	3
1	1	3
2	0	3
3	0	3

Counterexample path The actual record is E. Before the signal the chain below reaches a record where the proposition is false. Its length 2 is the first nesting depth that fails there.

Step	From	Shared view	To
1	E	1	F
2	F	2	B

Public signal The positive signal admits $\{D, E, F\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.19 K19. Three independent flags

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	1	0	+	−	1	−
C	2	0	1	+	−	1	−
D	3	1	1	+	+	2	∞
E	0	2	2	+	−	1	−
F	1	3	2	+	+	2	∞
G	2	2	3	+	+	2	∞
H	3	3	3	+	+	3	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	6	4
2	6	4
3	6	4

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	7	4
1	4	4
2	1	4
3	0	4
4	0	4

Counterexample path The actual record is H. Before the signal the chain below reaches a record where the proposition is false. Its length 3 is the first nesting depth that fails there.

Step	From	Shared view	To
1	H	1	D
2	D	2	B
3	B	3	A

Public signal The positive signal admits $\{D, F, G, H\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.20 K20. Three votes, quorum of two

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	1	0	−	−	0	−
C	2	0	1	−	+	0	0
D	3	1	1	+	+	1	1
E	0	2	2	−	+	0	0
F	1	3	2	+	+	1	1
G	2	2	3	+	+	1	1
H	3	3	3	+	+	2	2

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	2	3
2	2	3
3	2	2

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	4
1	1	1
2	0	0
3	0	0

Counterexample path The actual record is H. Before the signal the chain below reaches a record where the proposition is false. Its length 2 is the first nesting depth that fails there.

Step	From	Shared view	To
1	H	1	D
2	D	2	B

Public signal The positive signal admits $\{C, D, E, F, G, H\}$. A counterexample still

occurs at distance 2 after restricting to those records.

19.21 K21. Exactly one residual flag

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	1	0	+	+	1	∞
C	2	0	1	+	+	1	∞
D	0	2	2	+	−	1	−

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	2	2
2	2	2
3	2	2

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	3	2
1	0	2
2	0	2

Counterexample path The actual record is C. Before the signal the chain below reaches a record where the proposition is false. Its length 1 is the first nesting depth that fails there.

Step	From	Shared view	To
1	C	2	A

Public signal The positive signal admits $\{B, C\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.22 K22. Even parity announcement

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	1	1	+	−	∞	−
C	2	2	2	+	+	∞	∞
D	3	3	3	+	+	∞	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	3	2
2	3	2
3	3	2

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	3	2
1	3	2

Knowledge certificate Record D has no counterexample in its indistinguishability component even before the announcement.

Public signal The positive signal admits $\{C, D\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.23 K23. Odd parity announcement

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	+	−	∞	−
B	1	1	1	−	+	0	0
C	2	2	2	−	+	0	0
D	3	3	3	+	+	∞	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	2	1
2	2	1
3	2	1

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	2	1
1	2	1

Knowledge certificate Record D has no counterexample in its indistinguishability component even before the announcement.

Public signal The positive signal admits $\{B, C, D\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.24 K24. One-sided relay constraint

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	0	1	−	+	0	0
C	2	1	1	+	+	1	1
D	0	2	2	+	−	1	−
E	1	2	3	+	+	1	1
F	2	3	3	+	+	2	2

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	2	2
2	4	3
3	3	2

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in

each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	3
1	1	1
2	0	0
3	0	0

Counterexample path The actual record is F. Before the signal the chain below reaches a record where the proposition is false. Its length 2 is the first nesting depth that fails there.

Step	From	Shared view	To
1	F	1	C
2	C	3	B

Public signal The positive signal admits $\{B, C, E, F\}$. A counterexample still occurs at distance 2 after restricting to those records.

19.25 K25. Interlocked receipt flags

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	–	–	0	–
B	1	1	0	+	+	1	∞
C	2	1	1	+	+	2	∞
D	2	2	2	+	+	3	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	3	3
2	3	3
3	2	3

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	3	3
1	2	3
2	1	3
3	0	3
4	0	3

Counterexample path The actual record is D. Before the signal the chain below reaches a record where the proposition is false. Its length 3 is the first nesting depth that fails there.

Step	From	Shared view	To
1	D	1	C
2	C	2	B
3	B	3	A

Public signal The positive signal admits $\{B, C, D\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.26 K26. Three flags with an excluded triple

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity

means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	1	0	+	+	1	∞
C	2	0	1	+	−	1	−
D	3	1	1	+	+	2	∞
E	0	2	2	+	+	1	∞
F	1	3	2	+	+	2	∞
G	2	2	3	+	+	2	∞

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	5	5
2	5	5
3	5	5

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	6	5
1	3	5
2	0	5
3	0	5

Counterexample path The actual record is G. Before the signal the chain below reaches a record where the proposition is false. Its length 2 is the first nesting depth that fails there.

Step	From	Shared view	To
1	G	1	C
2	C	2	A

Public signal The positive signal admits $\{B, D, E, F, G\}$. Every run now connected to the actual record satisfies the proposition, so it is common knowledge there.

19.27 K27. Replicated majority evidence

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	+	0	0
B	1	1	1	−	+	0	0
C	2	0	1	+	+	1	1
D	3	2	2	−	+	0	0
E	0	3	2	+	+	1	1
F	1	2	3	+	+	1	1
G	2	3	3	+	−	2	−

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	2	1
2	2	1
3	2	1

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	4	3
1	1	0
2	0	0
3	0	0

Counterexample path The actual record is F. Before the signal the chain below reaches a record where the proposition is

false. Its length 1 is the first nesting depth that fails there.

Step	From	Shared view	To
1	F	1	B

Public signal The positive signal admits $\{A, B, C, D, E, F\}$. A counterexample still occurs at distance 1 after restricting to those records.

19.28 K28. One omitted flag at each observer

Three recorded flags define the possible runs. Observer numbered 1, 2, or 3 sees the other two flags but not its own omitted flag. The table records the resulting equivalence classes and the proposition being tested.

Possible runs and observation classes Distance is the shortest indistinguishability path to a counterexample. Infinity means common knowledge. The final column uses the world after a positive public signal.

Run	V1	V2	V3	P	Signal	d	d^+
A	0	0	0	−	−	0	−
B	1	1	0	+	+	1	1
C	0	2	1	+	+	1	1
D	1	3	1	−	+	0	0
E	2	2	2	+	+	1	1
F	3	3	2	−	+	0	0

Individual knowledge The count before the signal uses all runs. The count after it uses only runs marked by the signal.

Observer	Before	After
1	1	2
2	3	3
3	0	1

Iterated knowledge The table gives the sizes of X_j up to the first fixed point in each universe. A repeated count is a fixed point because the sets are nested.

j	Before	After
0	3	3
1	0	0
2	0	0

Counterexample path The actual record is E. Before the signal the chain below reaches a record where the proposition is false. Its length 1 is the first nesting depth that fails there.

Step	From	Shared view	To
1	E	3	F

Public signal The positive signal admits $\{B, C, D, E, F\}$. A counterexample still occurs at distance 1 after restricting to those records.

20 Atlas of finite progress certificates

Every transition is permitted by the local rules. Labels a, b, c record classes of service events. A fairness profile requires the named classes to occur infinitely often. Every last state has a self-loop carrying all labels, so a path to completion gives a fair run. The noncompletion test is Theorem 12.1.

20.1 P01. 4-state restartable pipeline

A worker can wait in each unfinished state, advance one stage, or retry after the last preparation. The direct finish carrying label c and the retry carrying label b are different permitted next steps.

Transition graph The input state is A and the completed state is D. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	b
3	C	D	b
4	A	A	a
5	B	B	a
6	C	C	a
7	C	D	c
8	C	B	b
9	D	D	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
BC	yes	yes	ab

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	4
ab	$A \to B$	2,8,5
abc	–	–

Nonvacuity The path $A \to B \to C \to D$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.2 P02. 4-state pipeline with a durable frontier

A durable frontier removes the backward retry without removing waiting. Intermediate work advances the frontier by one. The live server can still wait indefinitely at an unfinished stage if the scheduler only supplies label a.

Transition graph The input state is A and the completed state is D. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	b
3	C	D	b
4	A	A	a
5	B	B	a
6	C	C	a
7	C	D	c
8	D	D	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
B	yes	yes	a
C	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	4
ab	–	–
abc	–	–

Nonvacuity The path $A \to B \to C \to D$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.3 P03. 5-state restartable pipeline

A worker can wait in each unfinished state, advance one stage, or retry after the last preparation. The direct finish carrying label c and the retry carrying label b are different permitted next steps.

Transition graph The input state is A and the completed state is E. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	*b*
2	B	C	*b*
3	C	D	*b*
4	D	E	*b*
5	A	A	*a*
6	B	B	*a*
7	C	C	*a*
8	D	D	*a*
9	D	E	*c*
10	D	B	*b*
11	E	E	*abc*

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	*a*
BCD	yes	yes	*ab*

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	5
ab	$A \to B$	2,3,10,6
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.4 P04. 5-state pipeline with a durable frontier

A durable frontier removes the backward retry without removing waiting. Intermediate work advances the frontier by one. The live server can still wait indefinitely at an unfinished stage if the scheduler only supplies label a.

Transition graph The input state is A and the completed state is E. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	*b*
2	B	C	*b*
3	C	D	*b*
4	D	E	*b*
5	A	A	*a*
6	B	B	*a*
7	C	C	*a*
8	D	D	*a*
9	D	E	*c*
10	E	E	*abc*

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	*a*
B	yes	yes	*a*
C	yes	yes	*a*
D	yes	yes	*a*

Noncompleting fair runs For each label requirement, the table gives a finite prefix

to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	5
ab	–	–
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.5 P05. 6-state restartable pipeline

A worker can wait in each unfinished state, advance one stage, or retry after the last preparation. The direct finish carrying label c and the retry carrying label b are different permitted next steps.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	b
3	C	D	b
4	D	E	b
5	E	F	b
6	A	A	a
7	B	B	a
8	C	C	a
9	D	D	a
10	E	E	a
11	E	F	c
12	E	B	b
13	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
$BCDE$	yes	yes	ab

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	6
ab	$A \to B$	2,3,4,12,7
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.6 P06. 6-state pipeline with a durable frontier

A durable frontier removes the backward retry without removing waiting. Intermediate work advances the frontier by one. The live server can still wait indefinitely at an unfinished stage if the scheduler only supplies label a.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	b
3	C	D	b
4	D	E	b
5	E	F	b
6	A	A	a
7	B	B	a
8	C	C	a
9	D	D	a
10	E	E	a
11	E	F	c
12	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
B	yes	yes	a
C	yes	yes	a
D	yes	yes	a
E	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	6
ab	–	–
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.7 P07. 7-state restartable pipeline

A worker can wait in each unfinished state, advance one stage, or retry after the last preparation. The direct finish carrying label c and the retry carrying label b are different permitted next steps.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	b
3	C	D	b
4	D	E	b
5	E	F	b
6	F	G	b
7	A	A	a
8	B	B	a
9	C	C	a
10	D	D	a
11	E	E	a
12	F	F	a
13	F	G	c
14	F	B	b
15	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
$BCDEF$	yes	yes	ab

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	7
ab	$A \to B$	2,3,4,5,14,8
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E \to F \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.8 P08. 7-state pipeline with a durable frontier

A durable frontier removes the backward retry without removing waiting. Intermediate work advances the frontier by one. The live server can still wait indefinitely at an unfinished stage if the scheduler only supplies label a.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	b
3	C	D	b
4	D	E	b
5	E	F	b
6	F	G	b
7	A	A	a
8	B	B	a
9	C	C	a
10	D	D	a
11	E	E	a
12	F	F	a
13	F	G	c
14	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
B	yes	yes	a
C	yes	yes	a
D	yes	yes	a
E	yes	yes	a
F	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	7
ab	–	–
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E \to F \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.9 P09. Two workers wait on one another

The initial acquisition can be performed by either worker. The two unfinished ownership states can transfer control back and forth. An independent cleanup transition opens the last state.

Transition graph The input state is A and the completed state is E. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	A	C	b
3	B	C	b
4	C	B	a
5	B	D	c
6	C	D	c
7	D	E	ab
8	E	E	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
BC	yes	yes	ab
D	yes	no	$\varnothing$

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to C$	4,3
ab	$A \to B$	3,4
abc	–	–

Nonvacuity The path $A \to B \to D \to E$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.10 P10. One stranded restart state

One disconnected restart mode would allow all labels forever. It matters that this mode is unreachable from the declared input. The remaining loop covers only the first two labels.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	b
3	C	D	a
4	D	C	b
5	D	F	c
6	B	F	ab
7	E	E	abc
8	C	C	a
9	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
B	yes	no	$\varnothing$
CD	yes	yes	ab
E	no	yes	abc

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B \to C$	3,4
ab	$A \to B \to C$	3,4
abc	–	–

Nonvacuity The path $A \to B \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting

run yields a universal fair guarantee over a nonempty family of fair runs.

20.11 P11. Every label on a different branch

No single edge is marked with all obligations. The recurrent part can visit different obligation events along a cycle. That permits a fair execution that never reaches the result.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	b
3	C	D	a
4	D	E	b
5	E	B	c
6	C	F	c
7	F	G	ab
8	E	G	a
9	F	F	b
10	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
$BCDE$	yes	yes	abc
F	yes	yes	b

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B \to C$	3,4,5,2
ab	$A \to B$	2,3,4,5
abc	$A \to B$	2,3,4,5

Nonvacuity The path $A \to B \to C \to F \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.12 P12. Preparation and timeout modes

The prepare and timeout loops are separated by a one-way boundary. Their combined label set includes all three labels, but no unfinished execution visits both loops infinitely often.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	b
3	C	B	a
4	C	D	c
5	D	E	a
6	E	D	c
7	D	F	b
8	E	F	c
9	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
BC	yes	yes	ab
DE	yes	yes	ac

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B \to C$	3,2
ab	$A \to B$	2,3
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.13 P13. Durable commit hand-off

The first recurrent region is a repeated prepare. The second region starts after a durable decision and retains no path to the first. Learning the decision requires leaving it.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	a
3	C	B	b
4	C	D	c
5	D	E	a
6	E	D	a
7	E	F	b
8	F	G	c
9	D	G	ab
10	F	F	a
11	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
BC	yes	yes	ab
DE	yes	yes	a
F	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B$	2,3
ab	$A \to B$	2,3
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.14 P14. All callbacks before a retry

The system can complete every class of callback and still retry from the start before reaching the commit. Counting callbacks alone therefore gives an insufficient progress assumption.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	b
3	C	D	c
4	D	A	a
5	D	E	b
6	E	F	c
7	E	E	a
8	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
$ABCD$	yes	yes	abc
E	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	1,2,3,4
ab	(empty)	1,2,3,4
abc	(empty)	1,2,3,4

Nonvacuity The path $A \to B \to C \to D \to E \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.15 P15. Rebalance with an absorbing epoch

A successful reconfiguration is one-way. An unfinished run can stay either in the original epoch or in the new epoch, with a separate label requirement for finishing the handoff.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	b
3	C	A	a
4	C	D	c
5	D	E	b
6	E	F	a
7	F	E	b
8	F	G	c
9	D	G	a
10	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
ABC	yes	yes	ab
D	yes	no	$\varnothing$
EF	yes	yes	ab

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	1,2,3
ab	(empty)	1,2,3
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.16 P16. Two fair modes with separate exits

Two recurrent components are reachable by different initial choices. Each also permits an exit to completion. Existence of an exit gives no guarantee that the exit is used.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	A	D	b
3	B	C	ab
4	C	B	c
5	C	G	a
6	D	E	b
7	E	F	c
8	F	D	a
9	F	G	c
10	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
BC	yes	yes	abc
DEF	yes	yes	abc

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B$	3,4
ab	$A \to B$	3,4
abc	$A \to B$	3,4

Nonvacuity The path $A \to B \to C \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.17 P17. Admission before a resource wait

The admission label c occurs before the wait begins. One occurrence does not discharge a recurrence condition. At a later stage c can recur without any b events.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	c
2	B	C	a
3	C	D	b
4	D	C	a
5	D	E	b
6	E	F	a
7	E	E	c
8	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
B	yes	no	$\varnothing$
CD	yes	yes	ab
E	yes	yes	c

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B \to C \to D$	4,3
ab	$A \to B \to C$	3,4
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.18 P18. Restart waits with a shared cleanup

A shared cleanup state can return control to the initial retry mode. The apparent local components then belong to one strongly connected region of the full state space.

Transition graph The input state is A and the completed state is G. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	a
2	B	C	b
3	C	A	c
4	B	D	a
5	D	E	b
6	E	D	b
7	E	F	c
8	F	G	a
9	F	B	c
10	G	G	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
$ABCDEF$	yes	yes	abc

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	1,2,3
ab	(empty)	1,2,3
abc	(empty)	1,2,3

Nonvacuity The path $A \to B \to D \to E \to F \to G$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.19 P19. Eventual delivery in one direction

A delivery event can move past the waiting loop but the waiting loop itself contains no deliveries. The stated fairness condition distinguishes continuous waiting from infinitely repeated delivery.

Transition graph The input state is A and the completed state is E. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	B	b
2	B	C	a
3	C	B	a
4	B	D	b
5	D	E	c
6	D	D	a
7	E	E	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	no	$\varnothing$
BC	yes	yes	a
D	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	$A \to B$	2,3
ab	–	–
abc	–	–

Nonvacuity The path $A \to B \to D \to E$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

20.20 P20. Progress through a quiet interval

Several legal transitions carry no required label. Their loop is an infinite honest behavior but cannot meet any nonempty recurrence requirement by itself.

Transition graph The input state is A and the completed state is F. A blank label set means an honest internal step with no discharge of a named service event.

Edge	From	To	Labels
1	A	A	a
2	A	B	b
3	B	C	$\varnothing$
4	C	D	$\varnothing$
5	D	C	$\varnothing$
6	D	E	c
7	E	F	ab
8	E	E	a
9	F	F	abc

Unfinished components Components are taken after removing the completed state. The last column lists labels on internal edges. A label occurring only on an exit cannot recur inside.

States	Reachable	Cycle	Labels
A	yes	yes	a
B	yes	no	$\varnothing$
CD	yes	yes	$\varnothing$
E	yes	yes	a

Noncompleting fair runs For each label requirement, the table gives a finite prefix to a recurrent state and the edges of a closed walk to repeat. A dash indicates that the component criterion excludes such a run.

Required	Prefix	Repeated edges
a	(empty)	1
ab	–	–
abc	–	–

Nonvacuity The path $A \to B \to C \to D \to E \to F$ reaches completion. Continuing with the completed-state loop is fair for all three profiles. Thus an excluded noncompleting run yields a universal fair guarantee over a nonempty family of fair runs.

21 Atlas of capacity and higher schedule cells

An active set is admissible if and only if it is contained in a maximal listed permission. Different completed subsets can support the same active set. Each case compares direct cubical boundary ranks with the induced-subcomplex formula of Theorem 12.2. The prime in a column is its coefficient characteristic, rather than a parameter of the network.

21.1 H01. 3 independent actions, capacity 1

At most 1 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through C. The maximal sets are A, B, C. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	8	0	1	1
1	12	7	5	5

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1
0	1	0
1	0	0
2	0	3
3	0	2

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	ABC	2

Result The Euler characteristic is -4. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.2 H02. 3 independent actions, capacity 2

At most 2 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through C. The maximal sets are AB, AC, BC. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	8	0	1	1
1	12	7	0	0
2	6	5	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	0	0
3	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	ABC	1

Result The Euler characteristic is 2. Both coefficient choices give the same Betti numbers in this case. The induced calculation

and the direct cube calculation also agree in every degree over the two-element field.

21.3 H03. 4 independent actions, capacity 2

At most 2 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through D. The maximal sets are AB, AC, BC, AD, BD, CD. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	16	0	1	1
1	32	15	0	0
2	24	17	7	7

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	0	0
3	0	0	4
4	0	0	3

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCD$	3

Result The Euler characteristic is 8. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.4 H04. 4 independent actions, capacity 3

At most 3 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through D. The maximal sets are ABC, ABD, ACD, BCD. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	16	0	1	1
1	32	15	0	0
2	24	17	0	0
3	8	7	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
3	$ABCD$	1

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.5 H05. 5 independent actions, capacity 1

At most 1 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through E. The maximal sets are A, B, C, D, E. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	49	49

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1
0	1	0
1	0	0
2	0	10
3	0	20
4	0	15
5	0	4

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	$ABCDE$	4

Result The Euler characteristic is -48. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.6 H06. 5 independent actions, capacity 2

At most 2 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through E. The maximal sets are AB, AC, BC, AD, BD, CD, AE, BE, CE, DE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	31	31

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	0	0
3	0	0	10
4	0	0	15
5	0	0	6

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCDE$	6

Result The Euler characteristic is 32. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.7 H07. 5 independent actions, capacity 3

At most 3 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through E. The maximal sets are ABC, ABD, ACD, BCD, ABE, ACE, BCE, ADE, BDE, CDE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	0	0
3	40	31	9	9

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	5
5	0	0	0	4

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
3	$ABCDE$	4

Result The Euler characteristic is -8. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.8 H08. 6 independent actions, capacity 2

At most 2 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through F. The maximal sets are AB, AC, BC, AD, BD, CD, AE, BE,

CE, DE, AF, BF, CF, DF, and 1 more. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	0	0
2	240	129	111	111

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	0	0
3	0	0	20
4	0	0	45
5	0	0	36
6	0	0	10

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCDEF$	10

Result The Euler characteristic is 112. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.9 H09. 6 independent actions, capacity 3

At most 3 actions can be in progress at once. Every completed subset is allowed. Thus the guard selects all faces up to the stated active-set size.

Maximal permissions The action labels are A through F. The maximal sets are ABC, ABD, ACD, BCD, ABE, ACE, BCE, ADE, BDE, CDE, ABF, ACF, BCF, ADF, and 6 more. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	0	0
2	240	129	0	0
3	160	111	49	49

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	0	0
4	0	0	0	15
5	0	0	0	24
6	0	0	0	10

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
3	$ABCDEF$	10

Result The Euler characteristic is -48. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.10 H10. 4-action adjacency path

Simultaneous actions must be pairwise neighbors on a path. Only adjacent pairs have a square. The path of action permissions is connected, but induced restrictions to nonadjacent actions can be disconnected.

Maximal permissions The action labels are A through D. The maximal sets are AB, BC, CD. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	16	0	1	1
1	32	15	5	5
2	12	12	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	3	0
3	0	2	0
4	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	AC	1

Result The Euler characteristic is -4. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.11 H11. 4-action adjacency ring

The two ends of the preceding kind of action path are also permitted together. This closes one cycle in the permission graph and changes the induced restrictions.

Maximal permissions The action labels are A through D. The maximal sets are AB, BC, AD, CD. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	16	0	1	1
1	32	15	2	2
2	16	15	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The

table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	2	0
3	0	0	0
4	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	AC	1
2	$ABCD$	1

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.12 H12. 5-action adjacency path

Simultaneous actions must be pairwise neighbors on a path. Only adjacent pairs have a square. The path of action permissions is connected, but induced restrictions to nonadjacent actions can be disconnected.

Maximal permissions The action labels are A through E. The maximal sets are AB, BC, CD, DE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	17	17
2	32	32	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	6	0
3	0	8	0
4	0	3	0
5	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	ACE	2

Result The Euler characteristic is -16. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.13 H13. 5-action adjacency ring

The two ends of the preceding kind of action path are also permitted together. This closes one cycle in the permission graph and changes the induced restrictions.

Maximal permissions The action labels are A through E. The maximal sets are AB, BC, CD, AE, DE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	10	10
2	40	39	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	5	0
3	0	5	0
4	0	0	0
5	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	AC	1
2	$ABCDE$	1

Result The Euler characteristic is -8. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.14 H14. 6-action adjacency path

Simultaneous actions must be pairwise neighbors on a path. Only adjacent pairs have a square. The path of action permissions is connected, but induced restrictions to nonadjacent actions can be disconnected.

Maximal permissions The action labels are A through F. The maximal sets are AB, BC, CD, DE, EF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	49	49
2	80	80	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	10	0
3	0	20	0
4	0	15	0
5	0	4	0
6	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	ACE	2

Result The Euler characteristic is -48. Both coefficient choices give the same Betti

numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.15 H15. 6-action adjacency ring

The two ends of the preceding kind of action path are also permitted together. This closes one cycle in the permission graph and changes the induced restrictions.

Maximal permissions The action labels are A through F. The maximal sets are AB, BC, CD, DE, AF, EF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	34	34
2	96	95	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	9	0
3	0	16	0
4	0	9	0
5	0	0	0
6	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	ACE	2
2	$ABCDEF$	1

Result The Euler characteristic is -32. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.16 H16. Two exclusive pools of sizes 2 and 2

Actions in the same pool are mutually exclusive. Actions from different pools may overlap. There is no three-action face, but the pair graph can have several cycles.

Maximal permissions The action labels are A through D. The maximal sets are AC, BC, AD, BD. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	16	0	1	1
1	32	15	2	2
2	16	15	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	2	0
3	0	0	0
4	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	AB	1
2	$ABCD$	1

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.17 H17. Two exclusive pools of sizes 2 and 3

Actions in the same pool are mutually exclusive. Actions from different pools may overlap. There is no three-action face, but the pair graph can have several cycles.

Maximal permissions The action labels are A through E. The maximal sets are AC, BC, AD, BD, AE, BE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	6	6
2	48	43	5	5

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	4	0
3	0	2	0
4	0	0	3
5	0	0	2

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	CDE	2
2	$ABCDE$	2

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.18 H18. Two exclusive pools of sizes 3 and 3

Actions in the same pool are mutually exclusive. Actions from different pools may overlap. There is no three-action face, but the pair graph can have several cycles.

Maximal permissions The action labels are A through F. The maximal sets are AD, BD, CD, AE, BE, CE, AF, BF, CF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	10	10
2	144	119	25	25

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2
0	1	0	0
1	0	0	0
2	0	6	0
3	0	4	0
4	0	0	9
5	0	0	12
6	0	0	4

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	ABC	2
2	$ABCDEF$	4

Result The Euler characteristic is 16. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.19 H19. Separate pool capacities 2 and 1

The first 3 actions use a pool of capacity 2 and the last 2 use one of capacity 1. Both limits apply independently. The cubical space is the product of the two corresponding capacity skeletons.

Maximal permissions The action labels are A through E. The maximal sets are ABD, ACD, BCD, ABE, ACE, BCE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	1	1
2	72	48	1	1
3	24	23	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	1	0	0
3	0	0	1	0
4	0	0	0	0
5	0	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	DE	1
2	ABC	1
3	$ABCDE$	1

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation

and the direct cube calculation also agree in every degree over the two-element field.

21.20 H20. Separate pool capacities 2 and 1

The first 3 actions use a pool of capacity 2 and the last 3 use one of capacity 1. Both limits apply independently. The cubical space is the product of the two corresponding capacity skeletons.

Maximal permissions The action labels are A through F. The maximal sets are ABD, ACD, BCD, ABE, ACE, BCE, ABF, ACF, BCF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	5	5
2	192	124	1	1
3	72	67	5	5

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	3	0	0
3	0	2	1	0
4	0	0	0	0
5	0	0	0	3
6	0	0	0	2

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	DEF	2
2	ABC	1
3	$ABCDEF$	2

Result The Euler characteristic is -8. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.21 H21. Separate pool capacities 2 and 2

The first 4 actions use a pool of capacity 2 and the last 2 use one of capacity 2. Both limits apply independently. The cubical space is the product of the two corresponding capacity skeletons.

Maximal permissions The action labels are A through F. The maximal sets are $ABEF$, $ACEF$, $BCEF$, $ADEF$, $BDEF$, $CDEF$. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	0	0
2	240	129	7	7
3	128	104	0	0
4	24	24	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3	H^4
0	1	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	4	0	0
4	0	0	3	0	0
5	0	0	0	0	0
6	0	0	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCD$	3

Result The Euler characteristic is 8. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.22 H22. Separate pool capacities 2 and 2

The first 4 actions use a pool of capacity 2 and the last 3 use one of capacity 2. Both limits apply independently. The cubical space is the product of the two corresponding capacity skeletons.

Maximal permissions The action labels are A through G. The maximal sets are $ABEF$, $ACEF$, $BCEF$, $ADEF$, $BDEF$, $CDEF$, $ABEG$, $ACEG$, $BCEG$, $ADEG$, $BDEG$, $CDEG$, $ABFG$, $ACFG$, and 4 more. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	128	0	1	1
1	448	127	0	0
2	672	321	8	8
3	480	343	0	0
4	144	137	7	7

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3	H^4
0	1	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	5	0	0
4	0	0	3	0	0
5	0	0	0	0	0
6	0	0	0	0	4
7	0	0	0	0	3

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCD$	3
4	$ABCDEFG$	3

Result The Euler characteristic is 16. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.23 H23. One unbounded action

Action E can overlap every allowed combination of the other four. Those four have capacity two. The execution space has an

extra interval factor, so the new action does not remove the older holes.

Maximal permissions The action labels are A through E. The maximal sets are ABE, ACE, BCE, ADE, BDE, CDE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	7	7
3	24	24	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	4	0
4	0	0	3	0
5	0	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCD$	3

Result The Euler characteristic is 8. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.24 H24. Two filled clusters with one bridge

Each three-action cluster permits its full triple. The two clusters share a single allowed cross-pair, which makes their permission complex connected but does not fill mixed higher faces.

Maximal permissions The action labels are A through F. The maximal sets are ABC, CD, DEF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	33	33
2	112	96	0	0
3	16	16	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	8	0	0
3	0	14	0	0
4	0	9	0	0
5	0	2	0	0
6	0	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	AD	1

Result The Euler characteristic is -32. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.25 H25. A triangulated permission sphere

There are three disjoint pairs of actions. One can choose at most one from each pair. The allowed triples form the boundary of an octahedron in the permission complex.

Maximal permissions The action labels are A through F. The maximal sets are ACE, BCE, ADE, BDE, ACF, BCF, ADF, BDF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	3	3
2	192	126	3	3
3	64	63	1	1

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	3	0	0
3	0	0	0	0
4	0	0	3	0
5	0	0	0	0
6	0	0	0	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
1	AB	1
2	$ABCD$	1
3	$ABCDEF$	1

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.26 H26. Tetrahedral shell with a free action

Among the first four actions every proper subset is allowed. The fifth action can overlap every allowed subset. This adds an interval factor to the cubical boundary of the four-action cube.

Maximal permissions The action labels are A through E. The maximal sets are $ABCE$, $ABDE$, $ACDE$, $BCDE$. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	0	0
3	40	31	1	1
4	8	8	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3	H^4
0	1	0	0	0	0
1	0	0	0	0	0
2	0	0	0	0	0
3	0	0	0	0	0
4	0	0	0	1	0
5	0	0	0	0	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
3	$ABCD$	1

Result The Euler characteristic is 0. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.27 H27. Five triangular permissions

A consecutive triple on a cyclic list of five actions is allowed. Every pair occurs somewhere in a triple, but only the listed triples are simultaneous permissions.

Maximal permissions The action labels are A through E. The maximal sets are ABC, BCD, ABE, ADE, CDE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	11	11
3	20	20	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	5	0
4	0	0	5	0
5	0	0	1	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	ABD	1

Result The Euler characteristic is 12. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.28 H28. Projective-plane permission complex

Each allowed triangle has an explicit local authorization. The resulting permission complex is a triangulation of the real projective plane. Its mod-two and mod-three cohomology differ, which can be detected in the execution space.

Maximal permissions The action labels are A through F. The maximal sets are ABC, ABD, ACE, BDE, CDE, BCF, ADF, CDF, AEF, BEF. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	64	0	1	1
1	192	63	0	0
2	240	129	32	31
3	80	79	1	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	10	0
4	0	0	15	0
5	0	0	6	0
6	0	0	1	1

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	ACD	1
3	$ABCDEF$	1

Result The Euler characteristic is 32. The two characteristics give different Betti numbers. The resulting difference detects coefficient-sensitive topology of the permission complex, which a count of allowed cubes alone cannot detect.

21.29 H29. All pairs with one extra triple

Every pair is permitted. Only the first three actions can all overlap. The cubic comparison for that triple removes one relation among square boundaries.

Maximal permissions The action labels are A through E. The maximal sets are ABC, AD, BD, CD, AE, BE, CE, DE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	27	27
3	4	4	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	9	0
4	0	0	13	0
5	0	0	5	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCDE$	5

Result The Euler characteristic is 28. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

21.30 H30. All pairs with two extra triples

Two triples overlap in one action. The remaining pairs are allowed separately, with no further triple authorization.

Maximal permissions The action labels are A through E. The maximal sets are ABC, AD, BD, AE, BE, CDE. Subsets of a permitted active set are always permitted.

Direct cubical computation The boundary column counts the rank of the map out of the indicated degree, computed in $\mathbb{F}_2$. The two final columns give Betti numbers.

k	Cubes	Boundary	$\beta_k^{(2)}$	$\beta_k^{(3)}$
0	32	0	1	1
1	80	31	0	0
2	80	49	23	23
3	8	8	0	0

Induced permission subcomplexes For every subset of actions, compute reduced simplicial cohomology of the restriction. The table sums its contributions by subset size in characteristic two. A cubical degree k comes from reduced simplicial degree $k-1$.

$\lvert S\rvert$	H^0	H^1	H^2	H^3
0	1	0	0	0
1	0	0	0	0
2	0	0	0	0
3	0	0	8	0
4	0	0	11	0
5	0	0	4	0

One supporting subset per degree A nonzero row below identifies a restriction that contributes to the indicated cubical degree.

k	S	Contribution
0	$\varnothing$	1
2	$ABCDE$	4

Result The Euler characteristic is 24. Both coefficient choices give the same Betti numbers in this case. The induced calculation and the direct cube calculation also agree in every degree over the two-element field.

22 Using and extending the theory

22.1 Reading a completion failure

A service reports a request accepted, a replica reports its last durable version, and the monitoring layer records one delivery timeout. If the observations share a request identifier, they can be joined at that identifier. If a timeout has no bearing on durability, it belongs to the timing boundary. The honest record sheaf supplies the set or space of joint histories.

For an empty fiber, look for a contradiction certificate, such as a cycle, a cross-epoch mismatch, or an unsatisfied predicate at a single exposed transition. For a nonempty fiber, determine whether the boundary forces the property at issue. If it does not, the boundary kernel pair identifies the needed distinguishing data. If the boundary is feasible and the property is eventually to hold, examine the admission assumptions on future runs. The outstanding question can concern a possible delivery, a guarantee about all fair schedules, or a uniform latency bound.

22.2 Guide to the exact examples

Cases	Information calculated
C	Cuts, schedules, enabling sets and cube counts
Q	Minimal certificates, failure cuts and intersection
T	Tight deadlines, realizing times and negative cycles
A	Relative lifts, cokernels and Smith factors
G	Framed and unframed matches and automorphisms
R	Open paths and closed-loop multiplicities
K	Knowledge depths and counterexample paths
P	Reachable recurrence and completion certificates
H	Permitted cubes and induced cohomology

A ring with no fixed label in the relay atlas has empty completion in a set-valued interface. A nonzero holonomy in the gauge atlas can be allowed as a different global component until a face or boundary requires it to be trivial. An inconsistent face value in the abelian atlas is detected by a covector that annihilates all realizable values. These are realizations of a common question about a closed boundary. They differ in which paths are observable and what constitutes a witness.

The causal and capacity sections also have a common comparison. The entire event poset has a contractible cubical space when all enabled independence is genuine. Adding a resource guard can retain every total order while removing higher comparisons between them. The Betti numbers concern the guarded execution space. They cannot be deduced from the total-order count alone.

A finite transition graph can contain several reachable states from which completion is possible. The recurrence test identifies when there is still a fair run staying in the unfinished graph. A similar distinction for an infinite prefix tree is whether each finite prefix has a successful continuation, whether some fair continuation succeeds, or whether every fair run succeeds.

22.3 Boundary choices for further models

If an application needs trace data for nested transactions, a stage can expose a tree of descriptors with per-edge ports. Gluing remains a limit. An observation of the transaction outcome can be a dependent sum that forgets intermediate receipts. If an application needs real-time guarantees with bounded communication delays, it can extend the honest record by explicit clocks and feasible difference constraints. Each projection poses a concrete question about which witnesses and properties survive it.

Two open directions are directed equivalence for guarded cubical semantics and descent of fairness predicates under changing sites. The first asks when operations preserving reachable outcomes also preserve the spaces of directed ways to reach

them. The second asks when base change preserves the infinite limits used by a scheduler contract. The finite gluing and compactness statements give sufficient hypotheses in specific cases.

22.4 Reproducibility and disclosure

The companion source includes the LaTeX manuscript, exact case generators, and a machine-readable file containing the input data and intermediate certificates. The generators use integer, finite-field, graph, and finite-group arithmetic. They cross-check identities or explicit witnesses for each family, as described in the relevant section. Their output provides evidence for the specified finite instances. The general statements rest on the mathematical proofs in the text.

Generative AI assisted in preparing the manuscript, proofs, typesetting, and case generators. The byline is anonymous. The submission record can identify the responsible submitter in accordance with the journal’s requirements.

References

[1] Jacob Lurie. *Higher Topos Theory*. Annals of Mathematics Studies 170. Princeton University Press, 2009. arXiv:math/0608040.

[2] Jacob Lurie. *Kerodon*. Online reference, especially Chapters 4, 5, and 7. kerodon.net.

[3] The Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. Book and source.

[4] William G. Dwyer and Daniel M. Kan. Calculating simplicial localizations. *Journal of Pure and Applied Algebra* 18(1), 17–35, 1980. DOI.

[5] Rune Haugseng. Iterated spans and classical topological field theories. *Mathematische Zeitschrift* 289, 1427–1488, 2018. arXiv:1409.0837.

[6] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002. Author’s edition.

[7] Justin Curry. *Sheaves, Cosheaves and Applications*. Doctoral dissertation, University of Pennsylvania, 2014. arXiv:1303.3255.

[8] Jakob Hansen and Robert Ghrist. Toward a spectral theory of cellular sheaves. *Journal of Applied and Computational Topology* 3, 315–358, 2019. arXiv:1808.01513.

[9] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM* 21(7), 558–565, 1978. DOI.

[10] K. Mani Chandy and Leslie Lamport. Distributed snapshots: determining global states of distributed systems. *ACM Transactions on Computer Systems* 3(1), 63–75, 1985. Author’s copy.

[11] Friedemann Mattern. Virtual time and global states of distributed systems. In *Parallel and Distributed Algorithms*, 215–226. North-Holland, 1989. Full text.

[12] Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *Journal of the ACM* 46(6), 858–923, 1999. Full text.

[13] Lisbeth Fajstrup, Eric Goubault, and Martin Raussen. Algebraic topology and concurrency. *Theoretical Computer Science* 357(1–3), 241–278, 2006. Publisher.

[14] Nancy A. Lynch. *Distributed Algorithms*. Morgan Kaufmann, 1996. Author’s book page.

[15] Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM* 32(2), 374–382, 1985. Full text.

[16] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. *Journal of the ACM* 35(2), 288–323, 1988. MIT archive.

[17] Tushar Deepak Chandra and Sam Toueg. Unreliable failure detectors for reliable distributed systems. *Journal of the ACM* 43(2), 225–267, 1996. Cornell eCommons.

[18] Leslie Lamport. Paxos made simple. *ACM SIGACT News* 32(4), 51–58, 2001. Author's copy.

[19] Diego Ongaro and John Ousterhout. In search of an understandable consensus algorithm. In *USENIX Annual Technical Conference*, 305–319, 2014. Proceedings.

[20] David K. Gifford. Weighted voting for replicated data. In *Symposium on Operating Systems Principles*, 150–162, 1979. Full text.

[21] Maurice P. Herlihy and Jeannette M. Wing. Linearizability: a correctness condition for concurrent objects. *ACM Transactions on Programming Languages and Systems* 12(3), 463–492, 1990. Author-hosted text.

[22] Seth Gilbert and Nancy Lynch. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. *ACM SIGACT News* 33(2), 51–59, 2002. Full text.

[23] Joseph Y. Halpern and Yoram Moses. Knowledge and common knowledge in a distributed environment. *Journal of the ACM* 37(3), 549–587, 1990. arXiv:cs/0006009.

[24] Marc Shapiro, Nuno Preguiça, Carlos Baquero, and Marek Zawirski. A comprehensive study of convergent and commutative replicated data types. Inria Research Report RR-7506, 2011. Full text.

[25] Jim Gray and Leslie Lamport. Consensus on transaction commit. *ACM Transactions on Database Systems* 31(1), 133–160, 2006. arXiv:cs/0408036.

[26] Rina Dechter, Itay Meiri, and Judea Pearl. Temporal constraint networks. *Artificial Intelligence* 49(1–3), 61–95, 1991. DOI.

[27] Moshe Y. Vardi and Pierre Wolper. An automata-theoretic approach to automatic program verification. In *Symposium on Logic in Computer Science*, 332–344, 1986. Author's copy.

[28] Anthony Bahri, Martin Bendersky, Frederick R. Cohen, and Samuel Gitler. The polyhedral product functor: a method of decomposition for moment-angle complexes, arrangements and related spaces. *Advances in Mathematics* 225(3), 1634–1668, 2010. arXiv:0711.4689.

[29] Li Cai. On products in a real moment-angle manifold. *Journal of the Mathematical Society of Japan* 69(2), 503–528, 2017. arXiv:1410.5543.