Resource Acquisition in Higher Networks of Autonomous Agents

Effective protocols, finite observation, and non-computable homotopy types

Anonymous manuscript
10 November 2026

Abstract

Autonomous agents can negotiate for the computation, storage, and verification capacity they need for subsequent negotiations. We give such networks an effective operational semantics and a higher extension in which the type of agreement witnesses can have genuinely non-computable homotopy data. Open negotiations compose by homotopy pullback. Witness transport yields an exact interchange between composition and coherent inverse limits. We prove a sharp bound on homotopical observation: a span's inputs at degree $n+1$ determine its composite through degree n, and degree n can lose agreements. For each bit sequence b we build a connected space whose j-th homotopy group is trivial or cyclic of order two according to $b(j-1)$. Every Postnikov stage has a finite description with finitely many simplices in each degree. The Turing degree of b is the least oracle degree for a uniformly computable degreewise finite Kan presentation of the limit. A second family places the same obstruction in Postnikov extension classes while fixing all graded homotopy groups. For the halting sequence a computable stream of finite stages stabilizes at every fixed observation depth, while a computable stabilization bound would decide halting. This distinction gives a precise meaning to resource-enabled access. Examples cover escrow, compute allocation, coalition bargaining, and agreements with higher descent obstructions. A Lean HoTT companion formalizes structural proofs by path induction and equivalence construction.

Keywords autonomous agents, resource acquisition, homotopy type theory, spans, Postnikov towers, limit computability, negotiation.

Contents

1 Introduction **7**
1.1 Three independent indices 7
1.2 Main results and their status 8
1.3 Prior frameworks 9
1.4 Foundations and proof scope 10

I Effective negotiation **10**

2 Agents, resources, and local actions **10**
2.1 Holdings and endowments 10
2.2 Transitions, policies, and autonomy 12
2.3 Offers, authorities, and histories 13
2.4 An operational environment for negotiation 14
2.5 Resource accounting for computation itself 15

3 Settlement, conservation, and conditional progress **15**
3.1 A typed ledger 15
3.2 Atomic bundles and residual risk 17
3.3 What a fairness hypothesis buys 18
3.4 Composition of accounting claims 19

4 Effective presentations and their extension **19**
4.1 Three levels of presentation 19
4.2 Closure of the presented witness calculus 20
4.3 Constant towers, uniform towers, and ideal towers 22
4.4 Changing a representation 24

II Higher agreement **25**

5 The type of an agreement witness **25**
5.1 Identity types as reversible comparison 25
5.2 Transport in dependent contracts 26
5.3 Pullbacks as types of matches 26
5.4 Fibers, endpoints, and witness identity 27
5.5 Changing either leg by a homotopy 28

6 Open composition and network shapes **29**
6.1 The agreement at a shared boundary 29
6.2 Finite networks with many seams 30
6.3 Substitution and base change 32
6.4 Parallel composition and overlap 33

7 Coherent inverse limits of agreements **34**
7.1 An internal construction of the limit 34
7.2 Path and naturality formulas 35
7.3 Gluing commutes with realization 35
7.4 Dependent limits and lifting a fixed base point 37
7.5 Existence, choice, and convergence 37

8 Finite observation and the sharp dimension bound **38**
8.1 Truncation, connectivity, and observable equality 38
8.2 Why observation of a seam costs a dimension 38
8.3 Iterated pointed compatibility 40
8.4 The role of finite description 40

9 Finite probes and parametrized agreement **41**
9.1 A probe is a family of requests 41
9.2 Explicit finite tests on a Kan presentation 43
9.3 Combining probes and seams 44
9.4 An observation rule for contract verification 44

III Non-computable extensions **44**

10 An exact family of non-computable higher objects **45**
10.1 Finite descriptions in all positive dimensions 45
10.2 The split tower 46
10.3 Recovering homotopy invariants effectively 47
10.4 A weaker representation and a stronger example 48

11 Non-computability with fixed homotopy groups **49**
11.1 Two blocks with identical graded groups 49
11.2 A uniform diagnostic using a finite source 51
11.3 Assembling an infinite extension-class tower 52

12 Finite access and the cost of exactness **54**
12.1 Snapshots, hypotheses, and certificates 54
12.2 The halting family 54
12.3 A characterization by limit computability 56
12.4 Changing a budget is not restriction in depth 58
12.5 Resources as a cofinal access condition 58
12.6 Exact queries determined by finite information 59

13 Revisable observations and safe commitments **59**
13.1 Positive evidence in a changing display 59
13.2 A variance test for acceptance conditions 60
13.3 Settlement by versions 61
13.4 The order of two infinite operations 62
13.5 Finite distributed decisions 62

14 Pro-objects, realization, and the problem of choice **63**
14.1 A pro-object represents finite interaction 63
14.2 Realization preserves limits and can add maps 64
14.3 An effective tower can lack an effective coherent choice 65
14.4 Where finite connectedness helps 66

15 Composing canonical observation towers **67**
15.1 Levelwise gluing and the actual Postnikov tower 67
15.2 An effective implementation of the corrected tower 68
15.3 Canonical towers and maps in the ideal semantics 69
15.4 A change-of-context consequence 70

IV Negotiation and applications **70**

16 Strategies, payoffs, and approximate negotiation **71**
16.1 The strategic datum 71
16.2 Feasibility does not imply strategic stability 72
16.3 A finite horizon and a discounted continuation 73
16.4 Stability under payoff error 74
16.5 Changing what an agent can do 74
16.6 Payoffs that respect observation depth 75

17 Acquiring the means to observe **76**
17.1 A two-step purchase 76
17.2 Growing search service from negotiated surplus 77
17.3 No effective network extracts a stronger oracle from its own computation 79
17.4 Converting time and depth into one budget 79
17.5 Coalitions and the order of decisions 80

18 Local contracts and higher descent **81**
18.1 A site for contextual information 81
18.2 Descent of open negotiation 81
18.3 A discrete torsor with no global choice 82
18.4 A non-split extension as a coherence obstruction 83
18.5 A general obstruction and the space of resolutions 83
18.6 Restriction and realized observation 84

19 Resource grades and contextual refinement **85**
19.1 Charging a composite 85
19.2 Refinement through public interfaces 86
19.3 What tests can characterize a span? 87
19.4 A budgeted test is a joint problem 88
19.5 Limits of graded spans 88

20 Negotiation patterns **89**
20.1 Buying compute with a verification option 89
20.2 Negotiated visibility and a higher verification obstruction 90
20.3 A coalition building a finite observation 91
20.4 An alternating-offer benchmark 92
20.5 Observations that arrive after commitment 93
20.6 A conditional service guarantee 93

V Formal development **94**

21 Kernel-checked proofs in Lean HoTT **94**
21.1 The kernel and the library 94
21.2 The sequence of proof modules 94
21.3 How the checked gluing proof is assembled 95

21.4 The checked truncation comparison 96
21.5 A split tower in the kernel 97
21.6 Lifts through a Postnikov extension 98
21.7 Reproducibility and coverage 98

22 Scope and conclusions **99**
22.1 Three forms of adequacy 99
22.2 Semantic boundaries 99
22.3 Authorship and proof evidence 100

A Additional calculations and reference specifications **100**
A.1 A cycle with and without a coherence condition 100
A.2 Nonuniform snapshots and relative computation 101
A.3 An explicit finite calculation for a stage descriptor 102
A.4 A margin theorem for irreversible choices 103
A.5 A guide to the assumptions used in the main constructions . . . 103

1 Introduction

A network of agents changes its computational powers when it acquires resources. A buyer spends computation to search for a storage contract. A provider verifies that a deposit is funded before reserving capacity. An intermediary may negotiate separately for verification capacity before it can settle the buyer's contract. Each participant chooses its own offers, may walk away, and may use the acquired resources in later negotiations. An external semantics must describe the agreements that the independent choices jointly make possible. An operational semantics must specify what the participants can execute with the resources they actually have.

Agreements carry data beyond prices and allocations. Two verification plans can certify the same agreement by different means. A relabeling can transport a proof across a change in service identifier. Two such transports can themselves be related. In an open network the interfaces are supplied by other components, so these distinctions affect whether a local witness can be used in a larger negotiation. Homotopy type theory supplies a language in which the space of joint witnesses is defined by dependent sums and identity types. A path witnessing compatibility can affect the homotopy type of the whole agreement, even when each isolated offer has a unique witness. Sections 5 and 6 develop this point before taking any limits.

The model has an effective layer and an ideal layer. At the effective layer a bounded action of an agent is computable from local observations, current rights, and available fuel. The settlement machine enforces a resource invariant. Presentations of higher witnesses at a finite stage can be evaluated degree by degree. At the ideal layer the agreements can form spaces specified by arbitrary compatible towers. The same semantic composition acts on effective presentations and ideal towers. The comparison retains the visible behavior of effective objects and distinguishes operational evidence, such as fuel bounds and program codes, from the homotopy type obtained by forgetting that evidence.

1.1 Three independent indices

We use three indices because each controls a different source of information. An execution prefix has length t. It records which messages and effects have occurred. An observation has homotopical depth n. It records witnesses and their identifications up to a specified dimension. A computation stage s records how much search has been performed in an attempt to produce an accurate description of an observation. In the examples the canonical n-th ideal object exists before any execution or search has taken place. An evaluator given n and s can return a finite hypothesis about that object. Raising s revises hypotheses

at fixed n. Raising n reveals an additional part of the ideal object and can make earlier projection choices visible.

Physical holdings add a fourth parameter B, ordered by resource sufficiency. Holdings determine which finite pairs (n, s) a participant can afford. They need not be monotone along a real execution. A spend can lower a cash balance, an expiring reservation can remove a right, and a settlement can simultaneously decrease one coordinate and increase another. Whenever the model asserts eventual access, it states a cofinality or service assumption that connects actual holdings to the computational stages.

The choice to separate the indices has a precise mathematical consequence. The horizontal maps in an observation tower go from more detailed to less detailed data. The vertical transitions of a learner at fixed depth can change an earlier guess, and generally carry no compatibility map. The horizontal tower at each fixed computation stage in our principal example is coherent. A diagonal of the array of finite hypotheses generally fails to define a coherent tower. Section 12 gives an explicit instance and proves the corresponding stabilization results.

1.2 Main results and their status

The first result is an extension theorem. A finite effective presentation admits a constant tower. A uniformly presented tower admits an ordinary coherent tower by forgetting its algorithms. Constant towers preserve the homotopy pullbacks used for open composition. Uniform towers support degreewise effective composition through a path-object construction. Taking coherent limits commutes with the same composition. The paper uses the word extension in this sense. The forgetful map from operational programs to semantic spaces can identify programs that have different costs.

The second result quantifies the information needed to glue. Given $A \to B \leftarrow C$, the $(n+1)$-truncations of the three spaces and the induced legs determine the n-truncation of $A \times^h_B C$. One less dimension is insufficient, already for $A = C = \mathbf{1}$ and $B = K(\mathbf{C}_2, n+1)$. The exact recovery is a comparison of homotopy types, so it preserves witnesses and their identifications through dimension n. An equality of counts or feasible allocations is a weaker conclusion. The proof uses the identity type of a truncation and the stability of connected maps under base change.

The third result constructs non-computable higher objects explicitly. Write $A^b_e = \mathbf{C}_2$ when $b(e) = 1$ and $A^b_e = 0$ otherwise. Form the split Postnikov tower

$$X^b_n = \prod_{e<n} K(A^b_e, e+1), \qquad X^b = \prod_{e\in\mathbb{N}} K(A^b_e, e+1). \tag{1}$$

Each X^b_n is a finite observation in the sense specified below. The unmarked connected homotopy type of X^b already remembers b. Under a degreewise finite

Kan representation, an evaluator computes the orders of the pointed homotopy groups using data in consecutive simplicial degrees. An oracle for b gives such a representation. The exact oracle degree follows in both directions. This avoids making non-computability depend on a hidden labeling of the limit. A second family has the same graded homotopy groups for every bit sequence. Its extension classes are recovered by a finite coherent-lift test, so the same exact degree theorem holds with the groups fixed.

The fourth result analyzes access to that construction. If b is the halting sequence, a finite simulation produces a coherent tower of hypotheses at each stage s. Each fixed finite prefix of the tower stabilizes. A computable bound on its stabilization times would compute b. More generally, a computable approximation stabilizing pointwise exists exactly at the limit-computable level of this family. Larger oracle degrees still yield ideal objects all of whose stages individually have finite descriptions. Uniform access to those stages has the stronger degree requirement. These results separate existence of finite descriptions, computational access to guesses, and certification of exactness.

The fifth result is compositional and strategic. Feasibility descends and glues as a higher witness type. Payoff and best response require an additional strategy model, with deviations accounted for in a shared environment. A uniform payoff error ε contributes at most 2ε to regret when the feasible strategy sets are fixed. When they change, coverage of deviations has a separate cost. We prove both forms and give an example where agreement gluing succeeds and equilibrium fails. The mathematical treatment connects this distinction to the acquisition of better information and the cost of obtaining it.

1.3 Prior frameworks

Asynchronous automata provide an operational account of actions, locality, and fairness [25, 18]. Autonomous agents and automated negotiation supply the distinction between the policy of a participant and the protocol in which it acts [33, 16]. Bargaining theory treats strategic delay and outside options [26, 30]. Compositional game theory makes a different decomposition, using open games and their backward payoff context [9]. Our span semantics has a type of feasibility witnesses and a separate strategic layer, so a proof about gluing feasibility cannot by itself prove equilibrium.

The higher layer uses the interpretation of types as spaces and the synthetic calculus of identity, truncation, and equivalence [32, 17, 29]. Homotopy pullbacks have an explicit type-theoretic form. Ordinary directed histories can index such types externally, and synthetic directed types offer another possible ambient language [28]. Spans and cospans have developed into general accounts of compo-

sition in applied category theory [13, 2]. The present construction uses spans of witness spaces and glues by matching their common interface. The account of profinite spaces uses the established mapping-space calculus for pro-categories [1, 27].

The computability layer is external. It uses classical recursion theory, a concrete code convention for presentations, and a direct stabilization argument [31, 6, 4]. Hyland's effective topos is a model in which computability is internal to a category [15]. Our effective presentation category makes no identification with that topos. Keeping the computability convention external lets us compare models of HoTT, reason about oracle lower bounds, and still use constructive identity-type proofs for the structural results.

1.4 Foundations and proof scope

We use intensional dependent type theory with a universe of small types, function extensionality, and the truncations indicated in each result. Function extensionality is used for sections, products, and tower paths. Univalence supports interpreting equivalences as identifications of types when that interpretation is useful. It is unnecessary for many of the explicit equivalences. The external calculations in simplicial sets and computability use classical mathematics. Section 21 identifies which general structural proofs are mechanized using the dedicated Lean HoTT library and which simplicial, recursion-theoretic, and strategic arguments are given in the manuscript. A proof of a general equivalence is different evidence from checking bounded instances. The companion consists of theorem proofs.

Part I
Effective negotiation

2 Agents, resources, and local actions

2.1 Holdings and endowments

Let I be a finite set of agents. Fix a set R of resource kinds, finite for the effective models in this section. A resource amount is an element of the commutative monoid $M = \mathbb{N}^R$ under componentwise addition. Amounts may encode unit intervals of computation, memory reservations, transferable budget, or verifiable storage capacity. The interpretation of addition presupposes common units and compatible ownership. Distinct commitments to the same physical service cannot be counted as two available units. To model expiring services one can put the

relevant time window in the resource kind or attach the interval to a separate capability. We use that second presentation for the negotiation examples.

An account vector $h = (h_i)_{i \in I}$ represents spendable holdings. Write $h_i \geq a$ when each coordinate of a is covered. The formula $h_i - a$ is ordinary componentwise subtraction under that guard. A reservation holds an amount outside the spendable vector. A consumed account records retired units, so conservation can include destruction by use. For a derivation with external supply, an explicit input ledger records what the environment issues. This makes it possible to distinguish acquisition by transfer, acquisition by external provision, and a bookkeeping error.

The order of M is a feasibility order. An agent's preference can favor a bundle with fewer units in one coordinate if it gains enough in another. We reserve the notation $\preceq$ for comparison of resource requirements and use u_i for utility. In the main effective model balances and amounts are discrete. Divisible resources can be handled in a rational cone when the operations and comparisons stay computable. Statements using a natural-valued ranking assume the discrete model or an explicitly supplied discrete rank.

The linear book of amounts is distinct from knowledge. After a participant receives a proof of a public contract, copying that proof does not duplicate a physical reservation. Evidence lives in a type of witnesses, whereas rights are validated by the ledger. A deterministic verifier can authorize a payment given one accepted proof and can ignore later copies of that proof once the settlement identifier is closed. Section 3 proves the corresponding invariant.

Definition 2.1 (Requirement vector)**.** A requirement for a bounded local action is a vector $c \in M$. A state exposes a spendable amount h_i, a finite set of named capabilities with admissible use intervals, and an observation o_i. An action is feasible when $c \preceq h_i$ and its capability predicate holds at o_i. Resource vectors compose by addition for charges incurred sequentially. Requirements for reuse of an unconsumed capability compose by conjunction of its predicates.

Example 2.2 (The direction of the order)**.** An agent can own four compute quanta and a storage reservation, while a verifier asks for two compute quanta and a delegated proof-reading capability. The compute order verifies the first part of the requirement. The verifier cannot infer the second part from any increase in compute. A resource approximation theorem must either include capability access in its hypotheses or explicitly use a model in which the relevant capabilities are already granted. In Section 12 the observation evaluator runs under a stated access assumption.

2.2 Transitions, policies, and autonomy

A local state space L_i has a code representation. In the effective layer this representation supports the action of a deterministic controller on finite observations. A global state records the L_i, message buffers, the ledger, and an explicit environment state. The environment represents time or scheduling events that a participant does not choose. A transition label names the controller or environment component taking a step and gives the message or effect it proposes.

We specify a policy by a bounded partial evaluator with an explicit outcome for running out of fuel. Write

$$\mathrm{run}_i(o_i, \ell_i, m) \in \mathsf{Wait} + \mathsf{Timeout} + \sum_{a:\mathsf{Act}_i} L_i \times \mathbb{N},$$

where m is a computation bound and the final coordinate is a charge under the chosen machine model. For a fixed machine, simulation of at most m steps is total on the input representation. A proposed action enters the history only after the transition rule validates its resource and authority requirements against the current ledger. We use an explicit reservation or an atomic charge for the cost of the policy computation. A controller cannot perform an unaffordable search by declaring a timeout after spending more than the supplied fuel.

A nondeterministic or randomized policy may be represented by adding a finite choice input or a stream of random bits supplied as an environment input. For the qualitative transition semantics, retain the input used for the step. A probability model then puts a measure on such inputs and on the scheduler if needed. All safety statements below quantify over enabled histories and are independent of that measure. Claims about expected utility and discounting use a separate model.

We call the policy local if the same tuple (o_i, ℓ_i, m), with the same private choice inputs, produces the same result in every global state giving that tuple. This is a factorization through the observation map. Autonomous choice means each participant has its own policy and offer acceptance remains a transition under the participant's control or a delegated authority the participant selected. An external semantics may quantify over profiles of policies without giving the settlement machine a policy for choosing them.

Proposition 2.3 (Information indistinguishability)**.** *Fix a deterministic controller, its local state, and its fuel bound. If two global states give the same local observation and private inputs, the controller's proposed actions agree. If their ledgers differ on information needed by validation, one proposal can be enabled in the first state and disabled in the second.*

Proof. The two evaluations have equal inputs, so they follow the same bounded

computation. The validation relation is evaluated at the global state and can inspect the changed field. For a witness take a request to settle a reservation whose release message has reached the ledger in one state and has not yet reached the local observation. Keep the request, the observer's state, and the fuel fixed. The successful transition requires an open reservation and is disabled in the state where release has occurred. □

The second assertion is important for the distinction between a proposal and a commitment. A controller may send the same proposal twice after a delay. Settlement identifies the request by the reservation key and checks whether it remains open. Local feasibility estimates cannot replace that check.

2.3 Offers, authorities, and histories

An offer consists of a finite name, the parties, amounts promised and requested, a validity condition, and a verification rule. A service specification can be a finite code whose operational interpretation is a relation on verifiable outputs. The parties may attach distinct deadlines and cancellation rules. We treat eligibility, acceptance, reservation, and settlement as distinct events. A message that advertises a service supplies a possible transition at the receiver. It creates a ledger entry only after the protocol records consent and resource coverage.

A history is a composable list of transition witnesses. The endpoint map sends a history to its initial and terminal states. Empty histories and concatenation give an ordinary category if all transitions with the same label but different witnesses are retained as distinct generators and the only imposed equalities are the category axioms. Further identifications can come from a declared independence relation. A resource spend is directed. A renaming of equivalent witnesses is reversible. The free category of transitions and the witness space over a fixed transition boundary serve different purposes.

Let $x \xrightarrow{a} y$ and $x \xrightarrow{b} z$ be enabled steps. Call them strictly independent at x when each remains enabled after the other, their two composites reach the same state, and the observable output of each step is unchanged by exchanging their order. A sufficient condition is disjoint mutable footprints, no reads from the other's write footprint, and disjoint output channels or a declared commutation of channel operations. The phrase strictly independent includes commutation of time guards, which shared clock increments can invalidate. If repeated occurrences are tracked, the square interchanges the two named occurrences rather than identifying their labels globally.

Lemma 2.4 (Adjacent exchange)**.** *Suppose a history contains adjacent strictly independent occurrences, and the independence square preserves the boundary*

of the pair. Replacing the pair by the other side of the square gives an enabled history with the same outside states. Any additive cost on occurrences invariant under the square is also preserved.

Proof. The initial state of the adjacent pair is fixed. The independence square supplies two enabled steps with the same terminal state as the original pair. The suffix therefore has the same initial state as before. The unchanged prefix and suffix preserve their costs. Both sides of the square use the same two cost contributions, in exchanged order, and addition in M is commutative. □

The lemma generates equations on directed histories. If one wants to retain the reason for each exchange, the equations can instead be generating paths in a witness type over the two endpoints. Different sequences of exchanges are then paths between histories and may need higher identifications. A model that represents a schedule by its final balances loses this information. A model that regards a failed spend as an inverse of a successful spend changes the directed transition system. Our use of identity types retains exchanges of witnesses while keeping spend and refund as distinct forward events.

2.4 An operational environment for negotiation

The operational model assumes a state machine for each shared ledger. Several ledgers may communicate, but an atomic transition refers to one defined authority boundary. A history records the semantics of that state machine. It can be implemented by a single custodian or another system realizing the same transition relation. This paper uses the abstract relation and does not assume that arbitrary asynchronous parties can implement atomic broadcast with a specified failure tolerance.

A delivery action takes a message from a channel and hands it to the recipient. A timeout action uses a guard on the recorded time or logical round. Offers and acceptances may be reordered when their channels and guards permit it. The model therefore distinguishes the schedule on which a transaction completes from the set of histories on which it could complete. Resource safety is prefix closed. Eventual settlement is a liveness claim and requires assumptions on the schedule and the choices of the parties.

To expose an open component, choose visible input and output observations. Its behavior over these observations is a relation with witnesses. The witness remembers the local run, the chosen messages, and any certificate required by the interface. The definition leaves private states outside the boundary, although one can include them as part of a witness. Choosing what belongs in the boundary determines what can be matched when components are connected. For

example, exposing only the size of a requested capacity can admit a match across inconsistent validity intervals. An interface that carries the interval rejects that match by a failed identity condition.

2.5 Resource accounting for computation itself

The fuel for evaluating an offer is a resource distinct from the resource named in that offer. Let f denote a compute balance and let k_i be the number of charged steps. If evaluation finishes before its bound, the actual charge may be k_i. If it exhausts its bound m, a conservative ledger charge is m. Either rule satisfies the invariant that the participant spends at most what it reserved. The semantics of the verifier specifies which rule it uses, so payoff calculations know whether unused fuel is released.

A timeout witness certifies a fact about the execution of the bounded evaluator. It need not certify that an unbounded computation would diverge. In the halting construction, the corresponding finite report records that a machine has not halted by stage s. The report is exactly correct as a claim about the finite run. It is a revisable hypothesis when interpreted as a claim about the ideal space. This dual reading motivates the two interpretations of a finite observation developed later.

3 Settlement, conservation, and conditional progress

3.1 A typed ledger

Let J be a set of transaction names, with decidable equality in the effective implementation. Only finitely many names have a record in a finite operational state. A reservation is a record

$$q = (j, i, a, \rho, \theta, \zeta)$$

with name j, payer i, reserved amount a, a rule ρ assigning recipients and amounts, an expiry predicate θ, and a verification condition ζ. A rule may depend on a certificate. For each successful certificate w it produces allocations $a_k(w)$ and a retirement amount $d(w)$ satisfying $\sum_k a_k(w) + d(w) = a$. The ledger stores the rule chosen at opening. A later message cannot change the rule associated with an existing name. Closed names retain a status and enough information to identify duplicate requests. This is the abstract persistent state needed by the safety theorem.

For each name the possible statuses are unused, open, settled, and released. Opening requires an unused name, the appropriate authority, and a spendable

balance covering a. It subtracts a from the payer and creates the open record. Settlement consumes an open record, verifies an admissible witness under the stored rule and current guards, distributes $a_k(w)$, retires $d(w)$, and marks the name settled. Release consumes an open record under the cancellation or expiry rule, restores a to the original payer, and marks the name released. A query or duplicate acknowledgement has no resource effect. A distinct model may treat failed attempts as consuming computation, by giving that computation its own guarded transition.

The book includes a retirement balance $z \in M$. Let

$$\mathcal{T}(h, Q, z) = \sum_i h_i + \sum_{q \in Q} a(q) + z$$

where Q is the collection of open reservations. If explicit external supply has cumulative amount m, the expected total is $\mathcal{T}_0 + m$. Internal purchases, transfers, compute charges and expiry events can be handled entirely within this identity. A physical transfer across the accounting boundary is modeled as an input or a retirement, according to its direction.

Theorem 3.1 (Ledger safety)**.** *For every enabled finite history of the ledger, all balances remain in M, each reservation name is opened at most once and has at most one successful closing transition, and $\mathcal{T} = \mathcal{T}_0 + m$. In a closed system an increase in a participant's spendable holding is covered by a reduction elsewhere in the spendable or reserved accounts or by an explicit reversal of a previous reservation through release.*

Proof. Induct on the list of transitions. Initially the conclusions are part of the state specification. Opening uses a guard $a \preceq h_i$; the resulting holding $h_i - a$ lies in M. The held amount leaves the spendable sum and enters the reserved sum. A settlement is enabled only on an open record. Its allocations and retirement replace exactly the reserved amount by the rule's conservation identity. Release moves exactly the open amount to the payer. After either closing the name stays closed and cannot satisfy the guard for opening or successful closing. An external issue increases the holding sum and m by the same amount. A pure observation leaves all the quantities fixed. The invariant at the new endpoint follows in every case.

For the final assertion inspect the enabled transitions that can increase a spendable coordinate. When supply is fixed these are allocations from open reservations, releases, or direct transfers modeled by opening and closing together. All such increases are backed by the account whose value was reserved or debited. The sum identity gives the same constraint aggregated over each resource kind. □

Corollary 3.2 (Contended closing)**.** *In a linearized history, valid settlement and release requests that contend for the same open name have at most one successful closing effect, regardless of delivery order or duplication.*

Proof. The earlier successful request changes the status from open to a terminal status. Every later closing request requires open status. Duplicate requests are governed by the same condition. □

This corollary is about an authority boundary with a linearized ledger. The word linearized means that admissible effects admit an order modeled by a history, respecting the relevant ordering of the interface. Agents that see old snapshots can still emit competing requests. A commitment exists when a valid transition takes effect, and its acknowledgement may arrive later. The two events are separated in the operational semantics.

3.2 Atomic bundles and residual risk

A bilateral exchange can be specified by reserving the required debits of both parties and closing both reservations in one transition. Let $Q_0 \subseteq Q$ be a finite bundle of reservations controlled by a common settlement rule. Require all names in Q_0 to be open and the aggregate credit plus retirement to equal $\sum_{q \in Q_0} a(q)$. The same induction as in Theorem 3.1 proves conservation and at-most-once closing. In the successful transition the state changes on all named reservations at once. If a controller applies the same updates as individual transitions, an intervening release can make the second transition fail while the first remains effective.

Proposition 3.3 (Bundle linearization)**.** *Let Q_0 be an enabled bundle and suppose its single closing rule reads a fixed snapshot of all the named open records. Any order of applying the component balance additions to that snapshot, followed by marking all names closed in the same transition, gives the same terminal ledger. The analogous statement about separately guarded transitions can fail.*

Proof. The snapshot supplies each term once. Addition is associative and commutative, so the final balances and retirement are independent of the order of summation. Distinct names are marked closed once and have the same final status in all calculations. For the last assertion let two counterparties each reserve one unit and let the settlement transfer one reservation at a time. Between those transfers allow the second reservation to expire and be released. The one-sided terminal distribution is obtainable by the separate transitions and cannot arise from a successful atomic bundle close. □

The effect of atomicity is thus structural. A strategy claiming to exchange two resources can use an atomic bundle witness to prove absence of intermediate one-sided states at that interface. Across independent authority boundaries, a higher compatibility proof can express agreement between receipts. It cannot retroactively make two effects atomic. Later examples use compatibility witnesses for verification and a guarded ledger for spending.

3.3 What a fairness hypothesis buys

An open reservation can remain open when a required message is withheld, when a validation predicate never becomes true, or when an enabled request is never scheduled. Weak fairness for a named rule means that if it stays enabled at every state of some tail of the run, a step of that rule eventually occurs. If the rule can fire many times, the same condition applies again on subsequent tails. Weak fairness does not imply that a guard true only at isolated instants is eventually used.

Theorem 3.4 (Progress under persistent enabling)**.** *Fix a reservation name and a run of the ledger. Suppose the name opens at some time and after a finite prefix either has closed or has a closing request that remains enabled until it closes. If the scheduler is weakly fair for that request, the reservation eventually closes. If the reservation is still open at the start of that tail and exactly one terminal outcome is admissible until it closes, the closing transition has that outcome.*

Proof. Suppose it stays open forever. The first alternative in the hypothesis is excluded. The request is continuously enabled along the specified tail, so fairness yields a step of the request. That step closes the reservation, contradicting the assumption. In the second assertion the closing transition occurs on the specified tail and therefore has the unique admissible outcome. □

The outcome hypothesis is useful when settlement and release have complementary time guards. If release is the fallback after a deadline, and the clock eventually crosses the deadline and stays beyond it, the release request is continuously enabled for any still-open name. To force settlement one needs a persistent settlement guard, or a scheduler with a stronger timing guarantee over a finite window. Merely saying that the participants prefer settlement leaves this requirement open.

A natural-valued rank gives a second conditional progress theorem for negotiations with multiple outstanding tasks.

Proposition 3.5 (Progress with a finite descent rank)**.** *Suppose a phase has a rank $r \in \mathbb{N}$. Until a terminal state is reached, every transition is rank nonincreasing*

and, from any suffix that stays at a positive rank, some rank-decreasing transition eventually occurs. Assume rank zero is terminal. Then the phase terminates.

Proof. If it never terminated, the rank would be a nonincreasing sequence of positive integers. It can decrease at most its initial value minus one times, so there is a suffix of constant positive rank. The stipulated rank-decreasing transition on this suffix is a contradiction. □

The hypothesis can be verified by showing a descent rule remains enabled at each fixed rank and applying weak fairness. It can also follow from a bounded timer. The rank proof does not impose a preference for any particular agreement. An agent may end the phase by cancellation. Conversely, an agent that is free to issue counteroffers indefinitely need not admit such a rank. Negotiation liveness is therefore a theorem about a protocol and its scheduling and behavioral hypotheses.

3.4 Composition of accounting claims

A composite service may use a downstream verifier. Its local statement can have the form: if the verifier supplies a receipt within interval J, the provider can release an output within interval J'. A separate witness asserts the verifier’s own resource guards and availability. Matching the shared receipt type lets one apply both claims. The condition that the intervals actually match belongs to the boundary data. The condition that the verifier remains enabled belongs to a liveness argument. The condition that the same deposit is not used twice belongs to accounting.

The local assertions have different logical shapes. A receipt compatibility proof is a term of an identity type. Balance coverage is a proposition checked at a transition. Liveness quantifies over infinite schedules. Relating them requires a stated bridge. For example, an enabled settlement whose verification rule accepts every witness of a contract type converts an inhabitant of that type into an admissible proof input. It still requires an open deposit, a delivery step, and an enabling guard. The bridge will be used in Section 20, where higher witness spaces are part of executable negotiations.

4 Effective presentations and their extension

4.1 Three levels of presentation

An operational agent has a finite program and can be evaluated on finite data with finite fuel. The higher semantics of its agreement witnesses calls for a separate

presentation convention. We use Kan simplicial sets at that level. A finite description means a finite string interpreted by a fixed representation scheme. A degreewise finite presentation has finitely many simplices in each degree. A π-finite space has finitely many components, finite homotopy groups at every point, and is d-truncated for some finite d. These three notions have different purposes. A finite group G has a finite multiplication table. Its nerve BG has $|G|^k$ simplices in degree k. For nontrivial G the nerve can have nondegenerate simplices in arbitrarily high degrees despite presenting a 1-type.

Definition 4.1 (Degreewise finite Kan code)**.** A code for a Kan presentation is a collection of programs, finitely specified, that return the complete duplicate-free list S_k of names of k-simplices on input k and evaluate all face and degeneracy operations on valid names. The programs terminate on their specified inputs. Their values satisfy the simplicial identities, the lists present a Kan complex, and equality of names within a list is decidable. A morphism code evaluates a simplicial map on every valid input. A pointed code includes a vertex. We call a code a finite-stage code when its Kan complex also presents a π-finite space.

The totality and Kan conditions are semantic promises of this representation. An evaluator is entitled to use them on a valid code. There is no claim that arbitrary programs can be decided valid. For a fixed degree, a promised horn filler can be found by enumeration of the target degree and comparison of faces. A user of the code can stop at any fuel bound and return a timeout, or use a certified bound derived for a particular family. At the computational layer the listing program itself has a cost. At the semantic layer the code denotes a space even if one agent lacks that cost.

A morphism's code can be identified extensionally when it computes the same maps on all degrees. This quotient makes an ordinary category of presented objects and computable strict maps. It does not make equality of maps a decidable operation. The category of programs with their fuel costs retains intensional distinctions. A semantic map in spaces also forgets the presentation and can identify strict maps through homotopy. It can include maps with no computable representative on the given codes. The extension below retains effective input behavior through interpretation, without requiring semantic fullness of the program category.

4.2 Closure of the presented witness calculus

For a Kan complex B the evaluation map

$$B^{\Delta^1} \longrightarrow B \times B$$

is a Kan fibration, by the simplicial mapping property for the inclusion $\partial\Delta^1 \hookrightarrow \Delta^1$. Its diagonal section by constant paths is a weak equivalence from B. Hence a cospan of Kan complexes $A \xrightarrow{f} B \xleftarrow{g} C$ has a homotopy pullback represented by

$$(A \times C) \times_{B \times B} B^{\Delta^1}. \tag{2}$$

The displayed strict pullback is Kan because its map to $A \times C$ is a pullback of a fibration.

Lemma 4.2 (Computable path matching)**.** *Given degreewise finite Kan codes for A, B, C and computable simplicial legs f, g, Equation (2) has a degreewise finite Kan code, uniformly in the input codes. If the input spaces are π-finite, the result is π-finite.*

Proof. A k-simplex of B^{Δ^1} is a simplicial map $\Delta^k \times \Delta^1 \to B$. The source is the nerve of a finite poset, with finitely many nondegenerate simplices. A candidate map is specified by an assignment of a simplex of B to each nondegenerate simplex of the source, of the same degree. There are finitely many assignments. The simplicial identities for a map reduce to checking finitely many faces of these generators, expressing degenerate faces by degeneracies of their unique nondegenerate sources. This gives a duplicate-free enumeration of the maps. Evaluating on endpoints and precomposing with face or degeneracy maps are finite operations. Equation (2) now selects the triples whose endpoints match exactly the two images. The enumeration and all structure maps are uniform.

For the second assertion, the projection from the homotopy pullback to $A \times C$ has fibers path spaces between $f(a)$ and $g(c)$. Such a fiber is either empty or equivalent to a loop space of B. The long exact sequence of a fibration gives finite homotopy groups for the total space, since the relevant groups of the base and fibers are finite. In degree one this uses the group exact sequence and finiteness of an image and a kernel. The total space has finitely many components because finitely many base components each have a fiber with finitely many components and any point over the same base component can be transported to that fiber. Above the maximum of the truncation bounds for the base and fibers the groups vanish, giving finite truncation degree. This reasoning applies componentwise when the pullback is disconnected. □

A terminal code has one simplex in each degree. Products are obtained by listing pairs. The same path-object construction computes homotopy fibers and fiberwise intersection with another contract. Proof relevance affects the cost. An evaluator may need simplices of B through degree $k + 1$ to list the k-simplices of the path object. The dimension loss for truncations proved in Section 8 is a homotopy invariant form of this dependency.

Functoriality is strict in commuting maps of cospans. Send (a, c, p) to the images of a, c and postcompose p with the mediator map. The endpoint equations hold by strict commutativity. If the input square only commutes by a specified homotopy, the type-theoretic construction uses the square's paths to adjust the endpoints. A finite-stage effective implementation can require codes for those homotopies. The comparison theorem below uses strict maps for executable tower data and allows general homotopy coherence in the target.

Lemma 4.3 (Semantic closure of finite stages)**.** *The full subcategory of π-finite spaces admitting degreewise finite Kan codes is closed under finite limits in $\mathcal{S}$, even for cospans whose legs are initially given only as semantic maps.*

Proof. Let the mediator be presented by a d-truncated Kan complex B, choosing $d \geq 0$. Put $m = d + 1$ and replace B by its m-coskeleton $\mathrm{cosk}_m B$ [22]. This is the right Kan extension of its simplices through degree m, so a simplex in degree k consists of a compatible map from $\mathrm{sk}_m \Delta^k$ into B. These maps admit uniform finite lists by assigning images to the finitely many generators of the skeleton. The coskeleton agrees with B through degree m. It is Kan: horns through dimension $m + 1$ can use fillers in B, and in higher dimensions the m-skeleton of a horn already equals the m-skeleton of the simplex. Above degree $m - 1$ its homotopy groups vanish, since the full boundary of a simplex in dimension greater than m has that same m-skeleton. In degrees at most $m - 1$ the simplices and the elementary homotopies used to compute the groups agree with B. Components agree as well. Thus $B \to \mathrm{cosk}_m B$ is an equivalence.

Represent a leg into B by a simplicial map from its given Kan source, which is possible since every simplicial set is cofibrant and B is fibrant. Compose with the map to the coskeleton. By the coskeleton adjunction, the composite is determined by its table on simplices through degree m of the source. This table is finite, and once fixed it gives an algorithm on every degree of the source using restrictions to the m-skeleton. Thus each of the two semantic legs can be replaced by a computable leg to the new mediator. Apply Lemma 4.2. This proves closure under pullbacks, and the terminal code then gives closure under finite limits. The argument gives existence of codes from semantic maps, without a uniform procedure taking a semantic map with no representation as input. □

4.3 Constant towers, uniform towers, and ideal towers

Let $\mathcal{F} \subseteq \mathcal{S}$ be the full subcategory of π-finite spaces admitting a degreewise finite Kan code. Admitting a code means that some presentation exists. The space itself can have many codes. Let D denote the category whose objects are the

nonnegative integers and whose arrows are $m \to n$ for $m \geq n$. We write

$$\mathsf{Tow}(\mathcal{F}) = \mathrm{Fun}(D, \mathcal{F}).$$

An effective strict tower has a total algorithm for its stage codes and for the simplicial transition maps in each degree. More precisely, a single algorithm for (n, k) supplies the list and operations at level n, and a single algorithm supplies the adjacent maps on names. Composites define the remaining arrows. The uniformity is a condition on the tower presentation, stronger than separate effective presentability of each stage.

A diagram in $\mathsf{Tow}(\mathcal{F})$ can use non-computable choices of stages and maps. It can also be specified by homotopy-coherent maps, without choosing strict representatives. For the explicit examples strict towers suffice. Homotopical natural transformations between towers act as semantic refinements. Stagewise equivalences are the equivalences in the functor category. We keep the named depths, because resource observers ask questions at specified levels. Pro-objects will give a second category in which reindexing is allowed.

Definition 4.4 (Open negotiation semantics)**.** For a category of witness spaces with finite limits, the semantic data of an open component from U to V is a span $U \leftarrow W \rightarrow V$. Its apex W is the space of witnesses and its two maps expose the boundary observations. In the semantic span category the mapping space between endpoints is the core of the slice over $U \times V$, with composition by homotopy pullback. We apply this construction to spaces, finite observation spaces, and towers.

The use of the core makes equivalences of witness spans, with their endpoint homotopies, the invertible comparisons in each mapping space. A noninvertible map over the boundary can be retained as a refinement in an additional category of apex maps. This distinction avoids calling every map of presentations a path. It also keeps a fixed pair of endpoints separate from a choice of policy inside the component. The higher span construction supplies the coherences of composition [13, 14].

Theorem 4.5 (Semantic extension)**.** *The following constructions are compatible with open composition.*

1. *A presented finite-stage span determines a span in $\mathcal{F}$ and a constant span in $\mathsf{Tow}(\mathcal{F})$. The latter embedding of semantic spans is fully faithful.*

2. *A uniform strict tower of presented spans determines a span in $\mathsf{Tow}(\mathcal{F})$. Gluing uniform strict towers over a common presented mediator produces a uniform strict tower.*

3. *Taking a homotopy limit in $\mathcal{S}$ carries open tower composition to open composition of the limits.*

Proof. For the first assertion the interpretation of the strict path-object composite is the homotopy pullback by Lemma 4.2. The constant functor preserves finite limits, computed pointwise in the functor category. A natural transformation of two constant space diagrams is parametrized by the classifying space of D, which is contractible because 0 is terminal. Thus the constant functor is fully faithful. The same statement for maps over a fixed constant boundary follows by taking a homotopy fiber of the mapping-space restriction. Passing to cores yields fully faithful mapping spaces of spans.

For the second assertion use the algorithm in Lemma 4.2 at input (n, k). The transition to the preceding stage acts on a path-object witness by applying the three given maps as described above. Thus both the stage code and the transition code are uniform. Finite limits in a functor category are pointwise, so the strict construction represents the semantic composition.

For the last assertion, the limit functor on D-diagrams of spaces is right adjoint to the constant diagram functor and preserves limits. Since the embedding of $\mathcal{F}$ preserves finite limits by Lemma 4.3, it suffices to apply the limit functor to the pointwise pullback in spaces. Section 7 gives a direct synthetic proof for sequential towers and specifies what happens to a seam path. □

The theorem places the effective layer inside the ideal semantics with a precise meaning. Operational presentations retain actions, fuel and authority constraints. Their images under interpretation can have additional semantic refinements. Both layers use the same gluing law after interpretation. The ideal semantics permits an arbitrary diagram of finite observation spaces. Its limit can lie outside the category of computably presented spaces even when every stage lies in the effective stage category. Section 10 gives a family for which the gap is an arbitrary oracle degree.

4.4 Changing a representation

Two equivalent Kan presentations of one witness space may have different listing costs. An equivalence in the semantic category transports the existence and homotopy type of agreements. To transfer a resource guarantee one needs a computable translation with a cost bound. For example, converting an explicit list of every simplex to a succinct group multiplication table can change the cost of evaluating one face. Conversely, a succinct stage code may return a much larger complete list when queried in a high degree. Resource analysis must state which operation is being priced. The oracle lower bound of Section 10 is invariant

over all degreewise finite Kan codes satisfying the full-list promise. It imposes no uniform efficiency condition.

One can change to an enumeration interface that never signals completion. In that convention even a finite list may have an uncomputable last-arrival time as the degree varies. Section 10 gives a weaker arithmetic bound for enumerable Kan presentations. The distinction anticipates an analogous one for negotiations. A service may eventually report every valid offer, or report a complete list at a specified depth. The second service entails a completion certificate. An agent that can buy more execution time has more chances to find a valid offer. It does not automatically acquire the stronger service.

Part II
Higher agreement

5 The type of an agreement witness

5.1 Identity types as reversible comparison

Write $x =_A y$ for the identity type of points $x, y : A$. Its elements are paths and can themselves be compared by identity types. We write $p \cdot q$ for composition in the order in which the paths are traversed, p^{-1} for inversion, and $\mathrm{ap}(f, p)$ for application of a function to a path. The operations and their laws follow from the identity eliminator, denoted J. For example, fixing x, to construct a term depending on $y : A$ and $p : x = y$ it suffices to give it at $y = x, p = \mathrm{refl}_x$. Function extensionality identifies a path between functions with a family of pointwise paths. Dependent function extensionality applies to sections of a type family.

A protocol record can be regarded as a type and a witness as one term. A path between witnesses can express a permissible rearrangement, a reversible change of verification plan, or a relabeling. The interpretation specifies which comparisons are included. In a semantic construction a type may be supplied from the external model of spaces, with identity interpreted by homotopical path spaces. An operational controller does not acquire a procedure for deciding arbitrary identity types from that interpretation. Identity proves invariance under the reversible comparisons that the model includes.

A spent balance and an unspent balance are two different operational states even when a later refund can reach an equivalent state. A transition between states generally has no inverse path witnessing equality of states. We use the directed transition category for scheduling and irreversible effects. We use identity

types for the internal comparisons of witnesses over their endpoints. In a directed synthetic model this separation is expressed by two constructions, directed arrows and equivalences. The arguments here require only the witness types and ordinary indexing by directed histories.

5.2 Transport in dependent contracts

Let $C : B \to \mathcal{U}$ be a family of certificates over a service interface B. If $p : b = b'$, transport gives

$$\mathrm{transport}_p^C : C(b) \longrightarrow C(b').$$

Its inverse up to homotopy is transport along p^{-1}. Applying transport along $p \cdot q$ is homotopic to applying transport along p and then q, by induction on p, q. If B is a classifying space of a group, transport along loops may act nontrivially on $C(b)$. In that case the certificate is not determined by its endpoint labels. The path along which it was translated affects the result.

Lemma 5.1 (Paths in an agreement family)**.** *For $u : C(b)$ and $u' : C(b')$, there is an equivalence*

$$\big((b, u) =_{\sum_{x:B} C(x)} (b', u')\big) \simeq \sum_{p : b = b'} \big(\mathrm{transport}_p^C(u) = u'\big).$$

Proof. Path induction on an equality of total points defines the forward map, which sends reflexivity to $(\mathrm{refl}, \mathrm{refl})$. For the inverse first eliminate p and then eliminate the displayed equality in the fiber. The result in the two reflexive cases is reflexivity of the pair. Eliminating those same paths proves that both round trips are homotopic to the identities. These homotopies are terms, so they keep the information needed for comparisons between paths. □

This lemma explains why two readings of the same contract may be related even when their raw certificate terms differ. The term p translates the reading of the service identifier. The second component checks the translation against the target certificate. If one drops the second component, an invalid comparison can be recorded as valid. If one replaces the pair by a mere proposition of existence, the fact that there are multiple compatible translations becomes invisible.

5.3 Pullbacks as types of matches

Given functions $f : A \to B$ and $g : C \to B$, set

$$\mathsf{Match}(f, g) = \sum_{a : A} \sum_{c : C} \big(f(a) =_B g(c)\big). \tag{3}$$

We also write $A \times^h_B C$. A term is a pair of witnesses and a compatibility path. If B is a set, compatibility is a proposition, so a choice of equality proof carries no higher information. If B is a higher type, the choice of path may persist in the composite even when A, C are contractible.

Lemma 5.2 (Cone classification)**.** *For every type T there is a natural equivalence*

$$(T \to \mathsf{Match}(f, g)) \simeq \sum_{u:T\to A} \sum_{v:T\to C} \prod_{t:T} (f(u(t)) = g(v(t))).$$

Together with function extensionality this is the homotopy-pullback universal property.

Proof. Project a function w to its first two coordinates and project its last coordinate at every t. Conversely, pair the two functions pointwise with the selected path. The two composites agree component by component. The tuple identities and dependent function extensionality assemble the component identities into homotopies. Finally, function extensionality identifies the last product with the equality $f \circ u = g \circ v$ in $T \to B$. Precomposition in T commutes with the projections and the pairing, giving naturality. □

Lemma 5.3 (Comparing two matches)**.** *Let $w = (a, c, p)$ and $w' = (a', c', p')$ be matches. Their identity type is equivalent to*

$$\sum_{\alpha:a=a'} \sum_{\gamma:c=c'} (\mathrm{ap}(f, \alpha) \cdot p' = p \cdot \mathrm{ap}(g, \gamma)).$$

Proof. Apply Lemma 5.1 twice. Transport in the family of paths $f(a) = g(c)$ sends p to $\mathrm{ap}(f, \alpha)^{-1} \cdot p \cdot \mathrm{ap}(g, \gamma)$. This transport rule follows by induction on α, γ. Left multiplication by $\mathrm{ap}(f, \alpha)$ is an equivalence of the appropriate path types, with inverse given by the inverse path. It rewrites the resulting fiber equality as the square in the statement. Its validity and the endpoint identities follow from path composition laws. □

The equation in Lemma 5.3 is itself a path between paths in B. Comparing those squares retains another dimension. Thus the expression Match implements negotiation with certificates at arbitrary homotopy depth. Nothing in its definition sets a finite maximum depth. A representation may expose only $\|\mathsf{Match}\|_n$, but that is a subsequent observation of the already formed type.

5.4 Fibers, endpoints, and witness identity

For $f : A \to B$ and $b : B$, its homotopy fiber is

$$\mathrm{fib}_f(b) = \sum_{a:A} (f(a) = b).$$

It is a special case of matching against the one-point type. A path $b = b'$ transports the fiber at b to the fiber at b' by composing the second coordinate on the right. The proof uses J on the endpoint path. This expression is useful when an agent moves a commitment between equivalent service labels. The new commitment has a witness if and only if the old one does, and the equivalence describes how to carry the witness.

Proposition 5.4 (Reconstruction from fibers)**.** *There is an equivalence over B*

$$A \simeq \sum_{b:B} \mathrm{fib}_f(b), \qquad a \longmapsto (f(a), a, \mathrm{refl}).$$

Proof. The inverse projects to a. On a tuple (b, a, p), path induction on $p : f(a) = b$ gives a path from the image of the projected point to the original tuple. The other composite is the identity by definition. The forward map is over B strictly and the inverse is over B with the displayed path p. The round trips respect these identifications by the same path induction. □

Example 5.5 (A single component with distinct matches)**.** Let G be a nontrivial finite group, and use $B = BG$ with its distinguished point. Take $A = C = \mathbf{1}$ with constant legs. There is only one point in either outside type, and the match type is $\Omega BG \simeq G$ as a discrete space. Consequently a price and a chosen provider can be fixed while the interface still admits several matching receipts. If the mediator is replaced by $\tau_{\leq 0} BG \simeq \mathbf{1}$, the composite has just one matching receipt. The example detects the path information already at the level of distinct components of the composite.

Example 5.6 (A higher receipt)**.** Replace BG by $K(\mathbf{C}_2, 2)$. The same matching calculation gives $K(\mathbf{C}_2, 1)$, a connected type with a nontrivial group of self-comparisons. A statistic that counts connected components agrees with the statistic for a one-point match. A verifier that checks whether a comparison loop contracts can distinguish the two. Section 8 generalizes the calculation to every dimension.

5.5 Changing either leg by a homotopy

Suppose $f, f' : A \to B$, $g, g' : C \to B$, and $\alpha_a : f(a) = f'(a)$, $\gamma_c : g(c) = g'(c)$. A match (a, c, p) for f, g is carried to

$$(a, c, \alpha_a^{-1} \cdot p \cdot \gamma_c)$$

for f', g'. The inverse uses the inverse endpoint paths. The laws for paths give both round trips, and the outside projections are unchanged. Thus the match

type is invariant under a chosen comparison of each leg. Keeping the choice is important for naturality. Two distinct homotopies of the legs can give different transports of matches. For instance, a loop at the shared point of BG acts by multiplication on the group of matches of Example 5.5.

This invariance has an operational reading. A protocol adapter that changes which representative names appear can translate witnesses if it carries valid evidence for the two leg comparisons. An untrusted adapter supplying only an assertion that the names have the same display text does not define the transport. With the evidence, the construction is uniform in the witness and compatible with further adapters by the path laws.

6 Open composition and network shapes

6.1 The agreement at a shared boundary

An open component is specified by a span $U \xleftarrow{l} P \xrightarrow{r} V$. Two components with the same shared interface V have composite

$$\left(U \xleftarrow{l} P \xrightarrow{r} V\right) ; \left(V \xleftarrow{s} Q \xrightarrow{t} W\right) = \left(U \xleftarrow{l\pi_P} P \times^h_V Q \xrightarrow{t\pi_Q} W\right). \tag{4}$$

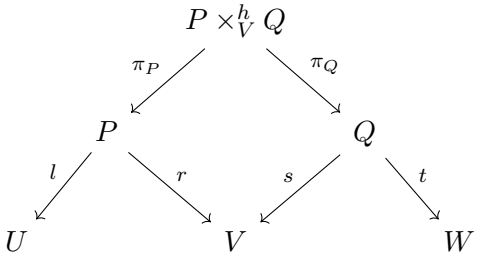


Figure 1: A composite witness projects to the two local witnesses. Its path in V compares the two middle legs while U, W remain available to the environment.

A point of the apex is a local choice from each component and a compatibility path in V. If the two witnesses depend on a particular schedule, those schedules or their exposed projections belong to the apices or the boundary. In the ordinary set case the apex is the usual relational join with witnesses. The homotopy type of the matching interface matters in the general case.

The two local choices need not be taken by agents acting in a central order. The order in (4) specifies the pattern of interfaces. The two ends of the composite may see a concurrent implementation of the joint witness. A theorem of compatibility

is about existence and transport of witnesses. Obtaining an enabled execution under a distributed scheduler also requires the scheduling assumptions of Section 3.

Proposition 6.1 (Units and associativity)**.** *The identity span $V \leftarrow V \rightarrow V$ acts as a two-sided unit up to equivalence of spans. Three composable spans have canonically equivalent bracketed composites, compatibly with the outside projections. The canonical associators satisfy the pentagon with a specified higher comparison.*

Proof. For the right unit, the extra witness is a point v with a path $r(p) = v$. Its type is contractible for each p. Projection to p is an equivalence and the endpoint homotopy is the stored path. The left unit uses the reversed path. In each case the contracting homotopy is J.

For associativity let the apices be P_1, P_2, P_3, with seams B_1, B_2 and legs $r_i : P_i \to B_i$, $l_{i+1} : P_{i+1} \to B_i$. A point in either bracketed composite can be flattened to

$$(p_1, p_2, p_3, \xi_1, \xi_2), \qquad \xi_i : r_i(p_i) = l_{i+1}(p_{i+1}).$$

Repacking is inverse by the path laws for tuples. The outside projections use only p_1, p_3 and are unchanged. For four factors choose the flattened witness type with three seam paths. Every edge of the pentagon is the equivalence obtained by flattening followed by repacking. The two routes around the pentagon agree up to the path obtained by cancelling adjacent repackings with flattenings. Equivalently, both maps are maps of limiting cones with the same designated comparison to the common cone. The space of such cone equivalences is contractible, which supplies the higher comparison and its coherent extensions. This argument uses the universal property in the category of cones. There can be nontrivial automorphisms of the apex considered without its cone. □

The last point is relevant when equivalent protocols have symmetric certificates. An arbitrary equivalence between the two bracketed apices may precompose the matching paths with a loop. The canonical associator is the one respecting all the projections and compatibility data. It is this choice that makes it legitimate to rebracket a network while transporting a specified settlement witness.

6.2 Finite networks with many seams

Take a finite directed multigraph $\Gamma = (V, E)$. For each vertex v let P_v be the space of local witnesses. For each edge $e : v \to w$ choose a mediator B_e and maps

$f_e : P_v \to B_e$, $g_e : P_w \to B_e$. The total witness type is

$$\mathsf{Agree}(\Gamma) = \sum_{p:\prod_{v\in V} P_v} \prod_{e:v\to w} \big(f_e(p_v) =_{B_e} g_e(p_w)\big). \tag{5}$$

This is a homotopy limit of the incidence diagram with an object B_e and two arrows into it for each edge. It is also the homotopy equalizer of two maps from $\prod_v P_v$ to $\prod_e B_e$. Self-edges use two maps from the same P_v. Two parallel edges impose separate seam witnesses. A triangle in the underlying graph does not impose a filled triangle of comparisons; such a condition must be added as data of the protocol.

Proposition 6.2 (Elimination along a cut)**.** *Partition the vertices into disjoint subsets V_0, V_1. Let P_i be the agreement type on the induced subgraph for V_i. If $J \subset E$ is the set of crossing edges, each P_i maps to $B_J = \prod_{e\in J} B_e$ by exposing its endpoint at each crossing edge. There is an equivalence*

$$\mathsf{Agree}(\Gamma) \simeq P_0 \times^h_{B_J} P_1.$$

For edges pointing from V_1 to V_0 the match path on the right is the inverse of the one on the left.

Proof. The vertex product splits into the two products on the partition. The edge conditions split into conditions internal to each side and conditions crossing it. Pair each internal condition with its own vertex data. For the crossing edges reverse the specified path when the second subset supplies the source. By function extensionality, the resulting product of paths is one path in B_J. This defines the forward map. The inverse projects its seam path to each B_e, reverses the designated ones, and assembles the products over the original graph. The two composites are homotopic to identities by the equivalence between paths in a product and products of paths, and by involutivity of path inversion. □

This cut calculation permits repeated elimination of local variables. If a cut has no crossing edges, $B_J = \mathbf{1}$ and the agreement type is the ordinary product of the two local agreement types. If one subgraph imposes a constraint on a resource held by the other, the shared ledger condition must be included in a crossing mediator or in a joint vertex. Declaring the product without that interface models a different network.

Example 6.3 (Contention carried by a seam)**.** Two independent search controllers can each produce a witness for a bid on a single slot. The product of the two witness types admits a pair of winning bids. A common settlement record can represent one open slot with a unique eligible recipient. Give each local witness

a map to that recipient interface, using distinct names for the two proposed winners. In the composite that purports to settle both as the unique winner, the seam path would identify different names in a discrete type. There is no such witness. The example eliminates the infeasible joint outcome at the level where the accounting constraint is exposed.

6.3 Substitution and base change

An interface map $u : U' \to U$ changes the visible input of a component by substitution. The new apex is $U' \times^h_U P$. This is the homotopy-theoretic counterpart of substituting names or restricting which counterparties may use a service. If a contract type is presented as a family $E : U \to \mathcal{U}$, the substituted family is $E \circ u$, and the two total-space descriptions agree by fiber reconstruction. There is a useful commuting law when substitution is applied before or after composition.

Theorem 6.4 (Substitution through a seam)**.** *For $u : U' \to U$, $l : P \to U$, $r : P \to V$, and $s : Q \to V$,*

$$U' \times^h_U (P \times^h_V Q) \simeq (U' \times^h_U P) \times^h_V Q.$$

The induced maps to U' and Q agree. The equivalence is natural in maps of the data and compatible with further substitutions.

Proof. An element on either side is equivalent to a tuple $(u', p, q, \alpha, \beta)$ with

$$\alpha : u(u') = l(p), \qquad \beta : r(p) = s(q).$$

The two repackings preserve all fields. The inverse laws use identities for dependent tuples, and naturality follows because the tuple maps use the same legs and preserve both seam paths. Equivalently both sides are limits of the diagram with the two seams U, V. Substitution by another map exposes a third seam and the same repacking uses all three paths. This gives the comparison with repeated substitution. □

For a square of interface maps, a chosen homotopy provides a transport between the two ways of substituting. If the square is a homotopy pullback, substitution commutes with taking dependent sums along the other leg. In internal type notation, take $p : E \to B$, $u : B' \to B$, and $C : E \to \mathcal{U}$. The fiber at b' of the substituted dependent sum has type

$$\sum_{e:E} \sum_{\eta : p(e) = u(b')} C(e).$$

The dependent sum along the pulled-back map has the same fields, with η as part of the pullback point. This verifies the needed Beck–Chevalley equivalence. The path orientation is fixed by the choice of fiber convention. This law is used when a downstream eligibility rule restricts a composed negotiation.

6.4 Parallel composition and overlap

The parallel product of two spans uses the product of the endpoints and of the apices. For two pairs of sequentially composable spans, the resulting sequential composite of products has an apex

$$(P_1 \times P_2) \times^{h}_{B_1 \times B_2} (Q_1 \times Q_2).$$

It is equivalent to $(P_1 \times^{h}_{B_1} Q_1) \times (P_2 \times^{h}_{B_2} Q_2)$. A compatibility path in the product is precisely a pair of paths, and the equivalence rearranges the components. This is the interchange law for independent wiring. The semantic tensor is the Cartesian product of interface spaces and implements parallel separation where there is no shared obligation.

A shared ledger can instead be carried as a base L. The product over L then imposes a matching ledger observation. If the components each use a read-only snapshot, matching that snapshot can be sufficient. If both consume resources from the same account, the ledger's transition relation or a splitting condition for the holdings must also be represented. Equality of initial snapshots does not ensure that two proposed withdrawals are jointly affordable. One way to express the splitting condition is to introduce an apex of admissible pairs (a_1, a_2) equipped with $a_1 + a_2 \preceq h$, with projection to the required amount of each component.

Lemma 6.5 (Separation by an explicit split)**.** *Let $h \in \mathbb{N}^R$ and suppose two components require $a_1, a_2 \in \mathbb{N}^R$, respectively. There is a disjoint earmarking of holdings h_1, h_2 with $a_i \preceq h_i$ and $h_1 + h_2 \preceq h$ if and only if $a_1 + a_2 \preceq h$. If each component's operational guards use only its own earmark and the independent transitions of Lemma 2.4, any interleaving of two enabled histories, preserving the local orders, is enabled under the combined earmark.*

Proof. Add the two inequalities for necessity. For sufficiency choose $h_i = a_i$. For the last assertion induct on the interleaving. The next step in either local history sees its local state and earmark unchanged by steps of the other history, except for commuting interface events allowed by the independence hypothesis. Thus the corresponding local guard and balance condition hold. The split prevents a debit against the other earmark. □

This result treats resources explicitly at the composition boundary. It prevents a semantic error that would otherwise look like a strategic opportunity. A policy cannot promise to use the same earmark simultaneously in both branches merely because it can realize each branch in isolation.

7 Coherent inverse limits of agreements

7.1 An internal construction of the limit

Work in dependent type theory with function extensionality and natural numbers. A sequential tower consists of types A_n and maps $r_n^A : A_{n+1} \to A_n$. The displayed data specify the chain, with maps for longer intervals obtained by composition. In spaces, the category of natural-number indices with descending arrows is freely generated by the adjacent arrows as an ordinary category. The corresponding homotopy-coherent diagram may be described using the chain of adjacent maps. A cone consists of compatible points and paths on adjacent edges. For composites, the cone paths are obtained by applying the transition maps and concatenating. This supplies the relations on longer arrows through the usual path coherences.

Define

$$L(A) = \sum_{x:\prod_{n:\mathbb{N}} A_n} \prod_{n:\mathbb{N}} (r_n^A(x_{n+1}) = x_n). \tag{6}$$

The equalities in this expression are identity types. It is the homotopy equalizer of the identity and the shift with restriction on $\prod_n A_n$. A map from T into $L(A)$ is equivalent to a family of maps $T \to A_n$ with adjacent compatibility homotopies. This follows by projecting and pairing pointwise, as in Lemma 5.2, and applying dependent function extensionality. It gives the universal property of the homotopy limit of the sequential diagram. One can also see the formula as the homotopy limit of a sequence modeled by a path-object replacement of the transition maps.

An element of $L(A)$ is a coherent choice, not merely a family of inhabitation proofs. In particular, a point x_n at each depth may be specified without any comparison of its projection to x_{n-1}. When the maps are restrictions of richer witnesses to poorer ones, a coherent element uses the same choices after transport at every step. A controller producing fresh guesses at each stage may fail to supply such a coherent sequence.

7.2 Path and naturality formulas

Lemma 7.1 (Path of a compatible family)**.** *Let* $(x, \xi), (y, \eta) : L(A)$*. Their identity type is equivalent to*

$$\sum_{q : \prod_n (x_n = y_n)} \prod_n (\mathrm{ap}(r_n^A, q_{n+1}) \cdot \eta_n = \xi_n \cdot q_n). \tag{7}$$

Proof. By Lemma 5.1 the identity comprises a path between x, y in $\prod_n A_n$ and an equality obtained after transporting ξ. Use function extensionality to replace the first path by q. Transport of the n-th component is

$$\mathrm{ap}(r_n^A, q_{n+1})^{-1} \cdot \xi_n \cdot q_n.$$

To verify it, start with a path between the two product sections and eliminate it by J. The pointwise paths are then reflexive, and both expressions are homotopic to ξ_n. Dependent function extensionality converts the transported equality of compatibility families into the pointwise equalities. Multiplying on the left by $\mathrm{ap}(r_n^A, q_{n+1})$ gives (7), with an inverse operation by the inverse path. Each step is an equivalence. □

A tower map $f : A \to B$ in this description consists of functions $f_n : A_n \to B_n$ and comparison paths

$$\theta_n^f(a) : r_n^B(f_{n+1}(a)) = f_n(r_n^A(a)).$$

The induced map on limits applies f_n to x_n and has compatibility path

$$\theta_n^f(x_{n+1}) \cdot \mathrm{ap}(f_n, \xi_n). \tag{8}$$

Composition of tower maps uses the composite of the comparison paths with the appropriate application of a function. Naturality of the induced maps follows by expansion and the functorial path laws. For strict presented tower maps the comparison paths are the canonical reflexive witnesses from the commuting squares in the model.

7.3 Gluing commutes with realization

Consider tower maps $f : A \to B$ and $g : C \to B$. At each level let $P_n = \mathsf{Match}(f_n, g_n)$. Its transition sends a point (a, c, p) to

$$\left(r_n^A a, r_n^C c,\ \theta_n^f(a)^{-1} \cdot \mathrm{ap}(r_n^B, p) \cdot \theta_n^g(c)\right). \tag{9}$$

The first comparison path is reversed because the new match starts at $f_n(r_n^A a)$. Formula (9) ends at $g_n(r_n^C c)$ as required.

Theorem 7.2 (Coherent gluing and limits)**.** *The natural comparison over $L(A)$ and $L(C)$ is an equivalence*

$$L(P) \simeq L(A) \times^{h}_{L(B)} L(C).$$

It carries the componentwise seam paths of a compatible family to the corresponding path between the two induced points of $L(B)$.

Proof. An element of $L(P)$ gives sequences a_n, c_n, p_n. Apply Lemma 5.3 to its comparison at the n-th step. The first two fields become paths

$$\xi_n : r^A_n(a_{n+1}) = a_n, \qquad \eta_n : r^C_n(c_{n+1}) = c_n.$$

The remaining square, writing $\theta^f_n = \theta^f_n(a_{n+1})$ and $\theta^g_n = \theta^g_n(c_{n+1})$, is

$$\mathrm{ap}(f_n, \xi_n) \cdot p_n = \big((\theta^f_n)^{-1} \cdot \mathrm{ap}(r^B_n, p_{n+1}) \cdot \theta^g_n\big) \cdot \mathrm{ap}(g_n, \eta_n). \tag{10}$$

Multiplication by θ^f_n and the path composition laws rewrite it as

$$\mathrm{ap}(r^B_n, p_{n+1}) \cdot (\theta^g_n \cdot \mathrm{ap}(g_n, \eta_n)) = (\theta^f_n \cdot \mathrm{ap}(f_n, \xi_n)) \cdot p_n.$$

The expression is exactly the n-th condition of Lemma 7.1 for a path from the image of (a, ξ) to the image of (c, η), using (8). Thus the data give a match in $L(B)$. Conversely unpack its path by Lemma 7.1, invert the indicated path multiplication, and use the inverse in Lemma 5.3. This recovers a term of $L(P)$. All rearrangements of products and sums here are the equivalences that project and reassemble dependent tuples. Their inverse laws, and the inverse laws of the two path lemmas, prove the equivalence. The two outside limit projections are unchanged. □

The theorem uses a higher proof even when there is only one connected component in each observed agreement. The equation comparing two paths remains an identity type and may have several terms or higher loops. This is how a compatibility of comparisons enters the limit. Replacing it by an informal assertion that the local receipts agree loses the content of the theorem.

Corollary 7.3 (Commutation for finite networks)**.** *For a finite graph of towers as in (5), the limit of the stagewise agreement tower is the agreement type of the limiting apices and mediators. It is natural in the outside projections, and hence compatible with grouping the graph into subcomponents.*

Proof. Sequential limits commute with finite products by the product projection and pairing equivalence. A homotopy equalizer is a pullback along a diagonal. Apply Theorem 7.2 to the product mediator and the relevant diagonal. Equivalently use the universal property of finite homotopy limits and the fact that L is a right adjoint on space-valued diagrams. The projections used in either proof are the projections from (5). □

7.4 Dependent limits and lifting a fixed base point

For a tower A and a family $C_n : A_n \to \mathcal{U}$, a transition in the displayed tower is a function

$$d_n(a) : C_{n+1}(a) \to C_n(r_n^A(a)).$$

It makes a tower of total types $T_n = \sum_{a:A_n} C_n(a)$. For a compatible base family $x = (a, \xi) : L(A)$, let

$$D(x) = \sum_{u:\prod_n C_n(a_n)} \prod_n (\mathrm{transport}_{\xi_n}^{C_n}(d_n(a_{n+1})(u_{n+1})) = u_n).$$

There is a natural equivalence $L(T) \simeq \sum_{x:L(A)} D(x)$ over $L(A)$. To see this, unpack each equality of total points using Lemma 5.1. Its first component supplies ξ_n and its second supplies the displayed equation in $D(x)$. Reassembly gives the inverse. The result shows exactly what a claimed limit refinement must lift. It must lift a specified coherent base family, with transport along its particular comparison paths.

7.5 Existence, choice, and convergence

A tower of nonempty spaces can have empty limit. In sets take $A_n = \{m \in \mathbb{N} \mid m \geq n\}$ and transitions the inclusions. Any compatible sequence would be a single integer at least every n. Finite nonempty stages change the conclusion under classical choice. For a tower with nonempty finite sets of components, those sets have a compatible choice by the finitely branching tree argument: the tree of compatible finite sequences has nodes at every height, obtained by projecting from the last stage. A branch gives compatible components. Choose one point of each chosen component and one connecting path for each transition. This yields an element of (6). The construction uses choice and a classical infinite-branch argument. It gives no program that finds such a branch from stage listings.

For arbitrary towers of π-finite stages, having finite stages does not imply the maps are truncation units. Conversely the Postnikov stages of a space need not be finite. A tower of observations and a Postnikov tower are kept distinct unless there is a stagewise identification. In spaces the Postnikov tower of a space converges to it. In a general ∞-topos the comparison with the limit of the Postnikov tower must be checked or assumed for the objects under consideration [23, 24]. The interchange Theorem 7.2 uses existing limits and does not require this separate convergence property.

8 Finite observation and the sharp dimension bound

8.1 Truncation, connectivity, and observable equality

For $n \geq 0$, an n-type has trivial homotopy above degree n. Synthetically, (-2)-types are contractible, (-1)-types are propositions, and an $(n+1)$-type has n-truncated identity types. The n-truncation $\|A\|_n$ is initial among maps from A to n-types. Its unit sends a to $|a|_n$. In the external category of spaces we use $\tau_{\leq n}A$ for the same construction. A map is n-connected when the n-truncation of each of its homotopy fibers is contractible. The unit $A \to \tau_{\leq n}A$ is n-connected. Truncation of a path type satisfies

$$\|(b = b')\|_n \;\simeq\; (|b|_{n+1} = |b'|_{n+1}). \tag{11}$$

This equivalence is induced by applying the $(n+1)$-truncation unit to paths. It is the identity-type theorem for truncation [32, Theorem 7.3.12]. The theorem holds without assuming the type A has finite homotopy groups. The distinction matters for comparing exact homotopical observations with finite encodings.

We use two elementary rules about connectivity. First, pullback of an n-connected map is n-connected, because its fibers are equivalent to the original fibers at the images of the base points. Second, an n-connected map induces an equivalence on n-truncations. One proof of the second rule writes $E \simeq \sum_{b:B} \mathrm{fib}_f(b)$. For an n-type Z, a map from each n-connected fiber to Z is equivalent to a point of Z by truncation elimination and contraction. Applying the equivalence fiber by fiber, maps $E \to Z$ are equivalent to maps $B \to Z$. The universal property of truncation implies the claim. In this argument the relevant equivalences are induced by precomposition, so the conclusion identifies the comparison map itself.

8.2 Why observation of a seam costs a dimension

Theorem 8.1 (One extra degree for a seam)**.** *For every $n \geq 0$ the natural map induced by truncation of a cospan gives an equivalence*

$$\tau_{\leq n}\big(A \times^h_B C\big) \;\simeq\; \tau_{\leq n}\left(\big(\tau_{\leq n+1}A\big) \times^h_{\tau_{\leq n+1}B} \big(\tau_{\leq n+1}C\big)\right).$$

The same n-truncated result can be computed with A, C unchanged and only the mediator truncated through $n+1$.

Proof. Set $B' = \tau_{\leq n+1}B$, $A' = \tau_{\leq n+1}A$, and $C' = \tau_{\leq n+1}C$. First compare $P = A \times^h_B C$ to $P_1 = A \times^h_{B'} C$ over $A \times C$. The map on fibers is the path map

$$(f(a) = g(c)) \longrightarrow (|f(a)|_{n+1} = |g(c)|_{n+1}).$$

By (11) it is equivalent to the n-truncation unit of the path type. In particular it is n-connected. For a map of dependent sums over the same base, the fiber over a total point is equivalent to the corresponding fiber map. This follows by eliminating the equality of base coordinates. Thus $P \to P_1$ is n-connected and gives an equivalence after $\tau_{\leq n}$.

Next P_1 is the base change of $P_2 = A' \times^h_{B'} C'$ along $A \times C \to A' \times C'$. The comparison is the base change because the endpoints factor through the two truncation units with the canonical unit homotopies. The product of the units is $(n+1)$-connected. Its fibers are products of the fibers of the units, and products of connected types in this range are connected, by the product rule for truncation or by two applications of its universal property. Base change preserves connectivity. In particular $P_1 \to P_2$ is n-connected and induces an equivalence on n-truncations. The composition gives the result. □

The statement allows the entire cospan to be observed at one common depth. The first part of the proof isolates the dimension spent on the compatibility path in B. The second part is a change of the base of a family of paths. It does not presume that the n-truncation preserves arbitrary pullbacks.

Proposition 8.2 (Sharpness with finite observations)**.** *For each $n \geq 0$ there are two cospans of π-finite spaces admitting degreewise finite Kan presentations whose $\tau_{\leq n}$-images are equivalent as diagrams and whose pullbacks have inequivalent n-truncations.*

Proof. Take $\mathbf{1} \to \mathbf{1} \leftarrow \mathbf{1}$ and $\mathbf{1} \to K(\mathbf{C}_2, n+1) \leftarrow \mathbf{1}$, with legs the basepoint. The mediators have contractible n-truncations, so the truncated diagrams are equivalent. The pullback of the first is $\mathbf{1}$. The pullback of the second is the pointed loop space $K(\mathbf{C}_2, n)$; for $n = 0$ this is the discrete group $\mathbf{C}_2$. Its n-truncation is itself and is not contractible. The finite presentations follow from the construction in Section 10. □

For a fixed graph of spans, the same overhead suffices simultaneously for all seams.

Corollary 8.3 (A graph needs one extra degree)**.** *The n-truncation of* $\mathsf{Agree}(\Gamma)$ *in* (5) *depends only on the graph data after $\tau_{\leq n+1}$. The bound does not grow with the length of a chain or the number of graph edges.*

Proof. Express the agreement as the pullback of the pair of maps $\prod_v P_v \to (\prod_e B_e) \times (\prod_e B_e)$ against the diagonal of $\prod_e B_e$. Truncation commutes with finite products in spaces. The truncated cospan in Theorem 8.1 is therefore determined by the vertex and edge spaces and maps truncated at $n+1$. Apply

the theorem once to the common mediator. Applying it separately after every composition can introduce additional apparent losses because that procedure discards information at intermediate seams. □

The graph result concerns the incidence diagram. If one adds compatibility constraints on paths between seams, those constraints introduce a second identity level. The dimension budget then depends on the specified diagram shape. For example, requiring a chosen path between two loop witnesses at a fixed basepoint in B uses an identity type in ΩB, and detecting a difference between such higher witnesses can require dimension $n + 2$ in B. The loop calculation below makes that precise.

8.3 Iterated pointed compatibility

Take a pointed type B and define $\Omega^0 B = B$, $\Omega^{d+1} B = \Omega(\Omega^d B)$, always pointed at reflexivity after the first step. By iterating (11) there is a canonical equivalence

$$\tau_{\leq n}(\Omega^d B) \simeq \Omega^d(\tau_{\leq n+d} B)$$

for $n \geq 0$. To prove it by induction, apply the formula for d at $n + 1$ and take the loop space, using the one-step path formula at the designated points of $\Omega^d B$. The canonical equivalences preserve these points because the maps are induced by truncation units. For $B = K(\mathbf{C}_2, n + d)$ with $d \geq 1$, the right side is $K(\mathbf{C}_2, n)$. The $(n + d - 1)$-truncation of B is contractible, so less depth can lose precisely that information.

In a protocol a d-fold pointed compatibility can arise when agents compare two verification plans, compare the translations of those plans, and repeat with the same designated references. The iterated loop model records the normalization in which all lower boundaries are fixed at their designated point. It is a local model for such coherence obligations. It does not identify every higher contract with an iterated loop space.

8.4 The role of finite description

Let a query be a map to an n-type Z. Any exact query $X \to Z$ factors through $\tau_{\leq n} X$, uniquely up to a contractible choice of factorization. It follows that an exact finite stage can suffice for a finite-valued predicate when that predicate is invariant under the higher paths. But factorization alone gives neither a finite description of $\tau_{\leq n} X$ nor an algorithm for evaluating its factor. If $X = K(\mathbb{Z}, 2)$, for example, $\tau_{\leq 2} X$ retains an infinite homotopy group. If X is in the bit family, the stage does admit a finite description, yet the uniform availability of exact stage codes is a computability question.

Finite-dimensional observation is also distinct from a finite simplicial skeleton. A skeleton includes only finitely many simplicial degrees and need not be Kan. A degreewise finite Kan code can answer a question about a skeleton of a given size, but the skeleton as a simplicial object does not usually represent the n-truncation. The sharpness proof is phrased for truncations because it compares homotopy-invariant observations. The effective recovery argument in the next section uses finitely many degrees of a Kan presentation for a different reason: the Kan condition reduces the homotopy equivalence relation in that query to an elementary filler.

9 Finite probes and parametrized agreement

9.1 A probe is a family of requests

A verification request can range over a space of parameters K. A provider then supplies a map $K \to B$, or a section of a family of admissible responses over K. If K is discrete and finite, the responses can be chosen independently once the local requirements are fixed. If K has paths, a response must also transport along them. A loop in K can impose a consistency condition on the entire family, even when every individual request can be satisfied. A finite CW complex or finite simplicial set can present such a parameter shape with finitely many generators and coherences.

A finite probe is distinct from a finite observation stage. A probe can be a sphere S^d, which has a finite CW presentation and is not a π-finite space for $d \geq 1$. An observation is a truncated witness space, often with finite homotopy groups. In this section the dimension of the probe is the highest cell dimension in its specified finite CW model. The dimension bounds below use that presentation. A lower-dimensional equivalent presentation, when available, gives a stronger bound.

We first prove a connectivity calculation in the external model of spaces. It is useful when a local witness is assigned to a family of related requests, and when a public protocol tests whether an entire family can be lifted through a verification interface.

Lemma 9.1 (Parametrized lifting)**.** *Let $p : E \to B$ be k-connected, with $k \geq 0$, and let K be a finite CW complex of dimension d. If $k \geq d$, the induced map*

$$\mathrm{Map}(K, E) \longrightarrow \mathrm{Map}(K, B)$$

is $(k - d)$-connected.

Proof. Represent p by a fibration and fix a map $g : K \to B$. The homotopy fiber of the displayed map over g is the space of sections of the pullback of the fibration along g. This follows either from the mapping-space universal property of a pullback or from the model by actual sections of a fibrant family. Each fiber of this family is k-connected.

A section exists by induction over the cells of K. At a j-cell the section given on its boundary is a sphere of dimension $j-1$ in the fiber, up to the homotopy trivialization over the cell. Since $j-1 \le d-1 \le k$, that boundary lifts to a section over the cell. At a zero-cell use nonemptiness of the fiber. In the fibration model the relative lift can be chosen with the specified boundary: a homotopy extension or the relative lifting property replaces the chosen extension by one agreeing with the prescribed boundary. Attaching the cells constructs a section.

To prove the higher connectivity, let $0 \le i \le k-d$. A map from S^i into the section space gives a section over $K \times S^i$ of the pullback of the original family along $K \times D^{i+1} \to K$. Extend it to $K \times D^{i+1}$. The relative pair has cells of dimension at most $d+i+1$, using the product CW structure and the relative cell of the disk. The same cell argument works because every required boundary has dimension at most $d+i \le k$. At $i=0$ this joins any two sections by a path. For positive i it contracts the represented sphere of sections, giving the vanishing of homotopy groups through $k-d$. Thus the section space is $(k-d)$-connected for every g. □

The proof uses cells to make explicit which coherence conditions are imposed. On a 1-cell one compares endpoint assignments by a path. On a 2-cell a chosen boundary loop must contract. More generally, the family of comparisons left by the boundary of a j-cell is a $(j-1)$-dimensional lifting problem. The extra parameter sphere for a homotopy group of sections increases the dimension of that boundary.

Theorem 9.2 (Observation depth for a finite probe)**.** *For $n \ge 0$ and a finite CW complex K of dimension d, the truncation unit induces an equivalence*

$$\tau_{\le n} \operatorname{Map}(K, X) \simeq \tau_{\le n} \operatorname{Map}(K, \tau_{\le n+d} X).$$

A corresponding statement holds for sections of a family over K, with $\tau_{\le n+d}$ applied fiberwise to the family.

Proof. The unit $X \to \tau_{\le n+d} X$ is $(n+d)$-connected. Apply Lemma 9.1 with $k = n+d$ and use the effect of an n-connected map on n-truncations. For the family statement consider the map from the original family to its fiberwise truncation in the slice over K. The homotopy fiber of the map of section spaces over a chosen truncated section is the space of sections of the pulled-back family

of fibers of the unit. These fibers are $(n + d)$-connected. The section-extension part of the preceding proof applies to that family and proves the result. □

Proposition 9.3 (Sharpness of the probe bound)**.** *For $d \geq 1$ and $n \geq 0$, the input degree $n+d$ in Theorem 9.2 cannot be uniformly replaced by $n+d-1$, even for finite observation spaces.*

Proof. Take $K = S^d$ and $X = K(\mathbf{C}_2, n + d)$, and compare it to the one-point space. Both input spaces become contractible at depth $n + d - 1$. For pointed maps the mapping type into X is $\Omega^d X \simeq K(\mathbf{C}_2, n)$, which is already n-truncated and noncontractible. For unpointed maps the same distinction holds. Evaluation at the basepoint gives the fibration over X with fiber $\mathrm{Map}_*(S^d, X)$. In the simplicial abelian group presentation of X, translation by the evaluation value identifies this fibration with $X \times \mathrm{Map}_*(S^d, X)$. The n-truncation of the first factor is contractible since $d \geq 1$. The second factor retains $K(\mathbf{C}_2, n)$. Both target spaces admit finite-stage codes. □

9.2 Explicit finite tests on a Kan presentation

Let K now be a simplicial set with finitely many nondegenerate simplices, all of dimension at most d. Let S have a degreewise finite Kan code. A j-simplex of $\mathrm{Map}(K, S)$ is a map $\Delta^j \times K \to S$. This source again has finitely many nondegenerate simplices, with dimension at most $j + d$. One way to verify the dimension bound is to represent a simplex of the product by degeneracies of nondegenerate simplices in the two factors. If it has dimension greater than the sum of their dimensions, the two degeneracy sets share an index, so the product simplex is degenerate. The finite assignments algorithm of Lemma 4.2 now supplies a complete list. Faces and degeneracies are computed by precomposition. Thus the mapping space has a degreewise finite Kan code, uniformly in the probe and target. The pointed mapping space has the same property, adding the fixed vertex condition.

For a given i the homotopy group order at a chosen map can be computed using the full lists of S through degree $i + d + 1$. The extra 1 is the degree used to compare two based representatives; it is not the homotopical observation depth in Theorem 9.2. The former is a bound on the input simplices for a particular finite calculation, while the latter is a homotopy-invariant equivalence. Both may matter for costing a verification request. Section 11 uses a finite probe to detect a Postnikov extension class that ordinary homotopy group queries miss.

9.3 Combining probes and seams

Suppose two providers expose witnesses in a common interface B and the customer requests a coherent family indexed by K. The total family is a point of $\mathrm{Map}(K, A \times^h_B C)$. Applying $\mathrm{Map}(K, -)$ to the pullback yields the equivalent type

$$\mathrm{Map}(K, A) \times^h_{\mathrm{Map}(K,B)} \mathrm{Map}(K, C).$$

The natural equivalence sends a family of pointwise seam paths to a path between the two interface maps, by function extensionality. Conversely, evaluate that path at each parameter. For a finite probe of dimension d, observing each of the input spaces through degree $n + d + 1$ suffices for the n-truncation of this family. To see it, first use Theorem 9.2 at degree $n + 1$ for the three mapping spaces. Then use Theorem 8.1 for the resulting cospan. Naturality of both comparisons ensures that the induced legs agree.

The bound is sharp in the worst case. Take the basepoint cospan with mediator $K(\mathbf{C}_2, n + d + 1)$ and compare to the point cospan. At input depth $n + d$ the diagrams agree after truncation. Their pullbacks are $K(\mathbf{C}_2, n + d)$ and a point. Use S^d for the probe when $d \geq 1$ and apply Proposition 9.3; for $d = 0$ use the one-point probe and Proposition 8.2. Hence the seam and the probe can each contribute a genuine part of the dimension requirement.

9.4 An observation rule for contract verification

Consider an eligibility rule that asks whether a particular map $K \to B$ lifts to E. The witness is a section, or equivalently a point in the homotopy fiber of $\mathrm{Map}(K, E) \to \mathrm{Map}(K, B)$ at that map. If the controller needs only inhabitation, the information requested is the propositional truncation of this fiber. The connectivity proofs have a version at this degree, with nonemptiness instead of a vanishing homotopy group. The cell argument says a family with $(d-1)$-connected fibers has a section over a d-dimensional finite probe. This is obtained from the first half of Lemma 9.1, and it also covers $d = 0$ when each fiber is inhabited. Requiring a path between possible sections is a stronger observation and uses the full d-connected hypothesis.

In a negotiation that delegates verification, the number of cells in a probe governs some implementation costs. Its dimension governs which layers of the mediator can affect the result. Two probes of the same dimension can differ greatly in the number of assignments the finite evaluator must consider. The manuscript keeps those bounds separate to avoid inferring a runtime guarantee from the homotopy dimension alone.

Part III
Non-computable extensions

10 An exact family of non-computable higher objects

10.1 Finite descriptions in all positive dimensions

Let A be a finite abelian group and $m \geq 1$. We use the simplicial abelian group

$$K(A,m)_k = Z^m(\Delta^k; A)$$

of normalized simplicial m-cocycles. A cochain is a function from the strictly increasing $(m+1)$-tuples of vertices to A. Its coboundary on an increasing $(m+2)$-tuple is the alternating sum of its values after deleting one vertex. A simplex is a cocycle if these sums vanish. A monotone vertex map acts by pullback, with value zero when the image of an increasing tuple is degenerate. This defines all faces and degeneracies. The code is uniform in a multiplication table for A, its additive inverse and identity, and m. A multiplication table for a finite group supplies all this information effectively. The cocycle constraints and the code validity are separate: the code enumerates a finite group of cochains, and the simplicial group structure ensures the Kan condition in all degrees.

By the Dold–Kan correspondence the normalized Moore complex of this simplicial abelian group is A concentrated in degree m. In the chain-map form of the correspondence, a map from the normalized chains of Δ^k into $A[m]$ is exactly an m-cocycle. Therefore the associated Kan space is an Eilenberg–Mac Lane space $K(A,m)$ and its only possibly nontrivial homotopy group is A in degree m [20, 11]. When $A = 0$ the code presents the point. When $k < m$ the group of m-cochains is zero, so there is one k-simplex even when $A \neq 0$. This last property makes it possible to give a degreewise finite model for an infinite product of such spaces.

Lemma 10.1 (Size of an explicit observation code)**.** *For $m \geq 1$ there are $|A|^{\binom{k}{m}}$ cocycles in $Z^m(\Delta^k; A)$, taking $\binom{k}{m} = 0$ when $k < m$.*

Proof. For $k < m$ there are no m-simplices. Otherwise restrict a cocycle to the m-simplices containing vertex 0. There are $\binom{k}{m}$ such simplices. The cocycle equation on the $(m+1)$-simplex consisting of 0 and any m-simplex not containing 0 determines its value from the restriction. Conversely give arbitrary values on the simplices containing 0. Take an $(m-1)$-cochain w with value on each tuple $j_1 < \cdots < j_m$ of nonzero vertices equal to the assigned value at $(0, j_1, \ldots, j_m)$, and value zero on tuples containing 0. Its coboundary agrees with the assigned

values because the only term that can be nonzero on an m-simplex containing 0 is the face omitting 0. Since the coboundary squares to zero, this yields the required cocycle. Thus the restriction is a bijection with $|A|^{\binom{k}{m}}$ choices. □

The size calculation concerns complete lists, not the size of a succinct stage descriptor. For a product having a nontrivial $\mathbf{C}_2$ factor in each degree $1, \dots, k$, the logarithm base two of the number of k-simplices is $\sum_{j=1}^{k} \binom{k}{j} = 2^k - 1$. The stage descriptor can just store k bits. A verifier that only reads which π_j are nontrivial can use that descriptor. A generic evaluator using the full-list interface may need far more computation. Our degree theorem is invariant under changes in running time of a valid full-list code.

10.2 The split tower

Fix an arbitrary bit sequence $b : \mathbb{N} \to \{0, 1\}$. Define $A^b_e = 0$ when $b(e) = 0$ and $A^b_e = \mathbf{C}_2$ when $b(e) = 1$. For $n \geq 0$ let

$$X^b_n = \prod_{0 \leq e < n} K(A^b_e, e + 1), \qquad X^b = \prod_{e \geq 0} K(A^b_e, e + 1).$$

The empty product X^b_0 is the point. The maps $r_n : X^b_{n+1} \to X^b_n$ forget the last factor and admit pointed sections by appending the zero cocycle. As maps of Kan complexes these projections are Kan fibrations. In each simplicial degree k, only factors with $e + 1 \leq k$ can contribute a nontrivial coordinate. Thus the infinite product is a degreewise finite Kan complex. Its product and simplicial operators are defined degreewise, with zeros supplied at newly visible coordinates. The finite products and the infinite product are also products in spaces, since all factors are Kan.

Theorem 10.2 (Reconstruction of the bit tower)**.** *The tower $X^b_\bullet$ is the Postnikov tower of X^b, with each stage in $\mathcal{F}$. For $j \geq 1$ the group $\pi_j(X^b)$ is A^b_{j-1}. The pointed and unpointed homotopy types therefore distinguish different bit sequences.*

Proof. Each finite product is connected and n-truncated, and its groups are finite. A stage code consists of the finite bit word $b|_n$ and the uniform cocycle programs. In the Kan presentation every factor has one vertex, so the full product is connected. The j-fold based loop functor commutes with products by its pullback definition. Homotopy groups of the product can also be computed using the degreewise product of normalized complexes in this particular simplicial abelian group. In degree j there is just one possibly nonzero normalized factor, so $\pi_j(X^b) = A^b_{j-1}$. This argument avoids any use of an infinite-product preservation rule for truncation. Projection $X^b \to X^b_n$ induces the isomorphisms in all degrees

$j \le n$, with connected source and target. It thus exhibits X^b_n as $\tau_{\le n}X^b$, by the universal property of Postnikov truncation or the n-connected fiber criterion. The comparison is compatible with restriction of depth. The inverse limit of the strict product-projection tower is its product of all factors, and it computes the homotopy limit because the projections are fibrations. Finally, any equivalence of the unpointed connected spaces induces isomorphisms of groups after choosing a path between the image of one point and the other. In particular it preserves the order of each group, which recovers $b(j-1)$. □

The stages are genuinely higher as soon as b has a 1 in a positive-index position. If b has infinitely many 1's, the object has nontrivial homotopy groups in unbounded dimensions. The construction uses split Postnikov data, with no nontrivial k-invariants. It locates non-computability already in a family of homotopy groups, before introducing possible additional sources of non-computability in actions or extension classes.

10.3 Recovering homotopy invariants effectively

Lemma 10.3 (Consecutive-degree recovery)**.** *For a pointed degreewise finite Kan code S and $j \ge 1$, the order $|\pi_j(S)|$ is computable uniformly from the code and j. A vertex can be chosen effectively when the complex is nonempty and no vertex has been supplied.*

Proof. List the j-simplices all of whose faces are the constant simplex at the chosen vertex. These are the based representatives of the j-th homotopy group. In a Kan complex, two such representatives σ, τ are equivalent when there is a $(j+1)$-simplex with last two faces σ, τ and other faces constant. This is the elementary based homotopy relation, with the convention that d_j and d_{j+1} are the two specified faces. The Kan condition implies that this elementary relation already gives the equivalence relation on these representatives [21, 8]. Enumerate the complete list in degree $j+1$, inspect the relevant faces, and form the equivalence classes of the based j-simplices. Their number is the group order. All lists terminate under the representation promises. If a point was not named, the complete list in degree zero supplies one. □

One can use the equivalent presentation of π_j by pointed homotopy classes of maps $\Delta^j/\partial\Delta^j \to S$. A representative is a based simplex, and the Kan horn operations reduce the homotopy cylinder to the elementary $(j+1)$-simplex condition. The hypothesis that the code presents a Kan complex is essential for this reduction. A list of simplices of an arbitrary nonfibrant model through degree $j+1$ does not in general suffice. Likewise a program that emits an enumeration

of degree $j+1$ without signaling its completion cannot decide that no elementary witness exists.

Theorem 10.4 (Exact oracle degree)**.** *For any oracle D and any b, the following conditions are equivalent.*

1. *b is computable relative to D.*
2. *The homotopy type X^b has a degreewise finite Kan code computable relative to D.*
3. *The named Postnikov stages of X^b have degreewise finite Kan codes uniformly computable relative to D.*

In condition 3 a uniform choice of stage maps can be required without changing the equivalence.

Proof. Under condition 1 use the uniform cocycle construction, reading only the bits with $e+1 \leq k$ to produce degree k of X^b. A face or degeneracy uses the same finite factor construction and supplies zero on factors that were trivial in the source degree. This yields condition 2. To obtain condition 3 under condition 1 use the first n bits to describe the stage and use projection as its transition map.

Under condition 2 compute $\left|\pi_{e+1}(X^b)\right|$ using Lemma 10.3, relative to D. By Theorem 10.2 the answer is 1 or 2 according to $b(e)$. This is a reduction $b \leq_T D$, independent of the presentation or the chosen vertex. Under condition 3 run the same computation at stage $e+1$; its group in that degree is already the final one. This proves condition 1. Each argument relativizes to the same oracle. The canonical construction in the first part supplies transition maps, proving the final claim. □

The theorem identifies the least Turing degree, since a code can be obtained using b as its oracle and any oracle supporting a code computes b. The non-computability resides in the homotopy type within the declared class, rather than merely in the length or format of its name. In particular an alternative degreewise finite Kan presentation, a different indexing of simplices, or an unpointed equivalence cannot make the non-computable family uniformly effective.

10.4 A weaker representation and a stronger example

Consider a weaker presentation of a pointed Kan complex. The simplices are enumerable in each degree, uniformly in the degree, with decidable equality on

names and computable face and degeneracy operations on valid names. Enumeration need not terminate. One can still recognize a positive elementary equivalence by finding a filler. Its absence is a universal condition on the enumeration. This yields an arithmetic upper bound on the possible bit invariants.

Proposition 10.5 (Enumerable Kan bound)**.** *If a connected X^b has an enumerable Kan presentation of this form, then $\{e : b(e) = 1\}$ is Σ^0_2. In particular for $b = \chi_{0^{(3)}}$, the characteristic function of the third Turing jump, there is no such presentation, although every Postnikov stage has a finite-stage code.*

Proof. Choose a vertex by waiting for the nonempty enumeration in degree zero, unless one is supplied. For $j = e + 1$, the group is nontrivial exactly when there is a based j-simplex with no elementary $(j + 1)$-simplex witnessing equivalence to the constant representative. The positive part includes a finite enumeration and computation witness for the simplex and its faces. The forbidden witness, if present, is detected at a finite enumeration and computation stage. Thus the predicate has form $\exists u \forall v\, R(e, u, v)$ with R recursive, using bounded simulations to incorporate computation times. This is a Σ^0_2 predicate. The third jump is Σ^0_3-complete and does not belong to Σ^0_2, by the strictness of the arithmetical hierarchy [6]. The finite-stage claim is Theorem 10.2. □

The proposition gives a second barrier that does not depend on full-list output. It concerns enumerable Kan presentations with the specified operations and does not classify arbitrary representations. For the halting sequence, the exact full-list barrier is already sufficient to demonstrate the separation. That sequence has the additional property that agents can generate better finite hypotheses without an oracle, as shown next.

11 Non-computability with fixed homotopy groups

The split tower proves a degree separation using the orders of homotopy groups. A finer construction leaves those orders, and indeed all the groups, fixed. It hides the bits in the maps that assemble two neighboring nonzero Postnikov layers. A finite probe recovers each bit, so the same oracle separation persists. This also gives an example where an agent with access to every group individually still lacks the information required to find a coherent lift.

11.1 Two blocks with identical graded groups

Use coefficients $\mathbb{F}_2 = \mathbf{C}_2$. For $m \geq 2$ let $K_m = K(\mathbf{C}_2, m)$ and let $u_m \in H^m(K_m; \mathbb{F}_2)$ be the universal class. Its cup square u_m^2 is nonzero. Indeed $B\mathbf{C}_2 =$

$\mathbb{RP}^\infty$ has cohomology ring $\mathbb{F}_2[t]$ with $\deg t = 1$. The class t^m is represented by a map $B\mathbf{C}_2 \to K_m$; pullback sends u_m^2 to $t^{2m} \neq 0$ [11, 12]. Define $k_m : K_m \to K_{2m}$ to be a pointed representative of u_m^2, and define

$$P_m^0 = \mathrm{fib}(0 : K_m \to K_{2m}), \qquad P_m^1 = \mathrm{fib}(k_m : K_m \to K_{2m}).$$

The fiber in both cases is taken at the distinguished point of K_{2m}, with a projection to K_m. Both projections have fiber K_{2m-1}. Both total spaces are connected and $(m-1)$-connected. By the homotopy long exact sequence, each total space has $\mathbf{C}_2$ in degrees $m, 2m-1$ and zero in every other positive degree. Because $m \geq 2$ the two degrees are distinct and both spaces are simply connected.

Proposition 11.1 (The space of lifts)**.** *Let $k : B \to Z$, $z : Z$, and $p : E \to B$ be the projection of $E = \sum_{b:B}(k(b) = z)$. For any $u : Q \to B$ there is an equivalence*

$$\sum_{v:Q\to E} (p \circ v \sim u) \;\simeq\; \prod_{q:Q} (k(u(q)) = z).$$

If p exhibits B as $\|E\|_m$, the right hand side for $u = \mathrm{id}_B$ is equivalent to the space of sections of the truncation unit of E. Consequently an equivalence of two total spaces with such projections induces an equivalence between their spaces of nullhomotopies, even when the equivalence is supplied without a comparison of the projections.

Proof. Regroup the lift v and its comparison homotopy into a family of pairs, one for each q. The pair is a point in the fiber of p over $u(q)$, which is equivalent to $k(u(q)) = z$. The forward map transports the path component of $v(q)$ along the displayed comparison to $u(q)$. The inverse takes $(u(q), \lambda_q)$ with reflexivity at $u(q)$, where $\lambda_q : k(u(q)) = z$. Path induction and function extensionality give the inverse homotopies. In particular the equivalence applies to paths between lifts as well as to their existence.

Write $\eta_E : E \to \|E\|_m$ for the unit and let $e : \|E\|_m \simeq B$ be the comparison with $e\eta_E \sim p$. The space of sections of η_E is equivalent to the section space of $e\eta_E$ by reindexing along e^{-1} and composing the section paths with e. The comparison homotopy identifies the latter with the section space of p. With $u = \mathrm{id}_B$ the first equivalence now identifies the section space with the displayed nullhomotopy space. This description of sections of η_E is intrinsic to E. In HoTT one can transport it along the universe path corresponding to a total-space equivalence by univalence. The Lean development constructs these equivalences in `Obstruction.hlean`. □

For P_m^0 the constant map has a reflexivity nullhomotopy. For P_m^1 a nullhomotopy would contradict the nonzero class u_m^2. The projections $P_m^i \to K_m$

exhibit their m-th Postnikov truncations, as follows from their groups and the projection's isomorphism on π_m. Thus Proposition 11.1 proves that the two total spaces are inequivalent. The same equivalence identifies the obstruction to a lift of a map $Q \to K_m$ with the obstruction to nullhomotoping its composite into K_{2m}.

Lemma 11.2 (Explicit Kan blocks)**.** *There are degreewise finite Kan presentations of P^i_m, uniform in m and $i \in \{0, 1\}$, with just the base simplex in every simplicial degree $j < m$.*

Proof. Use the cocycle presentation of K_m. The normalized cup product sends $x \in Z^m(\Delta^j; \mathbb{F}_2)$ to the cocycle $x \smile x$ in degree $2m$. On a strictly increasing list of $2m + 1$ vertices it evaluates x on the first $m + 1$ vertices and on the last $m + 1$ vertices, sharing the middle vertex, and multiplies the values. This product is natural under monotone vertex maps, including when a normalized face becomes degenerate. It represents u^2_m because the universal m-cocycle evaluates a simplicial m-simplex x on its full list of vertices. Its pullback along a simplex $x : \Delta^j \to K_m$ is precisely that cocycle x. Hence the pointwise cup square is a strict simplicial representative of k_m.

The simplicial abelian group E_{2m} of normalized $(2m - 1)$-cochains on Δ^j maps by the coboundary to K_{2m}. In the chain-map form of Dold–Kan, E_{2m} corresponds to the two-term complex with $\mathbb{F}_2$ in degrees $2m$ and $2m - 1$ and identity differential. A chain map into this complex is determined by the cochain in degree $2m - 1$, with its coboundary in degree $2m$. The coboundary map is induced by projection of the complex onto $\mathbb{F}_2[2m]$. Thus E_{2m} is contractible and the map is a Kan fibration, since the chain projection is surjective in positive degrees. It is a pointed contractible replacement of the basepoint map into K_{2m}.

The strict pullback along $x \mapsto i(x \smile x)$ therefore presents the indicated homotopy fiber. Its degree-j list consists of pairs (x, v) with x a normalized m-cocycle, v a normalized $(2m - 1)$-cochain, and

$$\delta v = i(x \smile x).$$

There are finitely many candidates and the equality is decidable, uniformly in the parameters. The total complex is Kan by base change of the fibration over the Kan complex K_m. When $j < m$, both cochains vanish because their degrees exceed j, so only the base simplex remains. □

11.2 A uniform diagnostic using a finite source

Let $Q_m = \mathrm{sk}_{2m} K_m$ in the cocycle model and let σ_m be the m-simplex with value 1 on the ordered full m-simplex. Its faces are constant, and it represents the

generator of $\pi_m(K_m)$. The skeleton has finitely many nondegenerate simplices, since its generators lie in finitely many degrees with finite lists. It has a single vertex. Restriction identifies $H^m(K_m;\mathbb{F}_2)$ with $H^m(Q_m;\mathbb{F}_2)$, because the cochains and coboundaries used in degree m are identical through degree $m+1$. This group has one nonzero element, by the Hurewicz theorem and the universal coefficient theorem for the $(m-1)$-connected K_m. The map on H^{2m} is injective: if a cocycle from K_m restricts to a coboundary on the skeleton, the same $(2m-1)$-cochain is defined on all of K_m and has that coboundary. Thus the square of the restricted universal class is nonzero.

Proposition 11.3 (The lift test)**.** *There is a pointed map $Q_m \to P^i_m$ sending σ_m to a nonzero class in $\pi_m(P^i_m)$ exactly when $i = 0$.*

Proof. For $i = 0$, compose the inclusion $Q_m \hookrightarrow K_m$ with the section. Its image has the desired class, since the projection induces an isomorphism on π_m. For $i = 1$, the composite $h : Q_m \to K_m$ of a proposed map with the projection would send σ_m to the nonzero class. In particular the class h^*u_m is nonzero. To see this directly, a zero class would make h nullhomotopic by representability of cohomology, hence trivial on the based sphere represented by σ_m. Since K_m is simply connected, a nullhomotopy of the underlying map also implies triviality on that pointed homotopy group. The unique nonzero H^m-class in Q_m has nonzero square. But the existence of the lift through P^1_m gives a nullhomotopy of $k_m \circ h$, whose represented class is $(h^*u_m)^2$. This is a contradiction. □

The test can be evaluated from any degreewise finite Kan code of the target block. Enumerate the pointed maps out of the finite probe by assigning images to its nondegenerate simplices and checking the faces, as in Section 9. For each map determine whether the image of the designated simplex represents a nonzero class using Lemma 10.3. All spaces are used through the relevant code interfaces. In particular a tester need not know a preferred projection of a given Kan model onto K_m. The proof of the test uses that projection to establish the invariant meaning of the answer.

11.3 Assembling an infinite extension-class tower

Let $m_e = 2^{e+1}$, and for a bit sequence b set

$$Y^b = \prod_{e \geq 0} P^{b(e)}_{m_e}.$$

The nonzero groups of a block lie at the two degrees m_e and $2m_e - 1$. The next block starts at $2m_e$. Thus the intervals are disjoint, and below any fixed degree only finitely many blocks contribute. Every Y^b is connected with the same

group in every positive degree, namely $\mathbf{C}_2$ in the named block degrees and zero elsewhere. The block models of Lemma 11.2 give a degreewise finite Kan model of Y^b.

The named Postnikov stage at depth n keeps all full blocks with $2m_e - 1 \leq n$ and keeps K_{m_e} for a block with $m_e \leq n < 2m_e - 1$. It discards the remaining blocks. The indicated maps from each full block to K_{m_e} induce the truncations, by the calculation of the homotopy groups. Their products have the desired truncation by the same bounded-degree argument as in Theorem 10.2. Finite-stage codes and transition maps are obtained from finitely many bits. A finite stage through the lower group of a block need not reveal its bit; the upper group and its connection to the lower one reveal it.

Theorem 11.4 (Exact degree of the extension-class family)**.** *For any oracle D, $b \leq_T D$ if and only if Y^b has a degreewise finite Kan presentation relative to D. Equivalently its named Postnikov stages admit a uniform relative presentation. The homotopy types Y^b distinguish the bit sequences although the graded homotopy groups are independent of b.*

Proof. An oracle computing the bits can list every required block and its maps, and in each simplicial degree only finitely many contribute. This gives the presentation of the product and of all named stages. Conversely fix e and let $m = m_e$. The projection $Y^b \to P_m^{b(e)}$ induces an isomorphism on π_m because the other factors have trivial groups there. A map from Q_m to the product that detects π_m projects to a map detecting π_m of this block. Conversely any detecting map into the block can be paired with the constant maps to the other factors. By Proposition 11.3 this existence query returns yes exactly when $b(e) = 0$. The finite test is effective relative to D for any presentation of the product. It works at any chosen vertex, because the space is connected and changing the vertex transports the detecting maps.

The same test works from a stage presentation of $\tau_{\leq 2m-1} Y^b$. This stage contains the full block and its projection, and all other factors present still have trivial π_m. This proves the equivalence with uniform named stages and the stated degree lower bound. If two products were equivalent, the finite-probe existence query, which is expressed in their pointed mapping spaces and homotopy groups, would have the same answer at each e. Thus the products distinguish the bit sequences. □

This construction isolates an information gap relevant to higher negotiations. A regulator can publish every local capability group, including all possible proof transformations at each degree, while withholding the extension class that controls whether those transformations lift coherently. The lift test measures the missing

datum. The result is not a promise of practical runtime: even the finite probe can require many explicit assignments. It is a theorem about which information can be recovered under the representation promises, and which oracle degree is necessary uniformly over the family.

12 Finite access and the cost of exactness

12.1 Snapshots, hypotheses, and certificates

Fix the space X^b and the designated Postnikov levels of Section 10. An *exact snapshot* at depth n is a finite-stage code together with the semantic fact that it presents X^b_n. A *hypothesis* is a stage code produced by a specified algorithm from n and a search parameter s. It is eventually correct when it presents X^b_n for all sufficiently large s at each fixed n. A *certificate of exactness* is evidence checked by a specified verifier that the code is exact. Existence of an exact snapshot for each n, a uniform generator for hypotheses, and an effective method to recognize exactness are distinct claims.

A fixed finite word $b|_n$ has a finite code for any sequence b, even when the sequence has no uniform program. The phrase there exists a code is a mathematical assertion with one existential quantifier per depth. An agent needs either a way to find that code, an access right to a service that supplies it, or a justified method of revising a hypothesis. More execution time strengthens the last method under a limit-computability assumption. Time by itself does not supply new non-computable input.

12.2 The halting family

Let $(M_e)_{e\in\mathbb{N}}$ be an effective enumeration of partial programs with a designated diagonal input e. Write

$$H(e) = \begin{cases} 1 & \text{if } M_e(e) \text{ halts,} \\ 0 & \text{otherwise,} \end{cases} \qquad h_s(e) = \begin{cases} 1 & \text{if it halts within } s \text{ steps,} \\ 0 & \text{otherwise.} \end{cases}$$

The function $(s, e) \mapsto h_s(e)$ is computable by bounded simulation. At any stage s form the tower $X^{h_s}_\bullet$. It has the canonical projection maps, so every fixed-stage row of the array is an effective coherent tower. For each n let s_n be the maximum of the halting times of those $e < n$ for which $H(e) = 1$, with maximum zero for the empty list. Then $X^{h_s}_n$ is the canonical X^H_n for all $s \geq s_n$. We use equality of the prefix words to give equality of the chosen descriptions. No search for an equivalence of spaces is needed.

Theorem 12.1 (Eventual access without an effective modulus)**.** *The halting tower has a computable family of finite hypotheses, coherently restricted in depth at each search stage and eventually exact at every fixed depth. There is no computable function $m(n)$ such that the stage-s hypothesis is exact for every $s \geq m(n)$ and every n. The same impossibility holds for a computable choice of one exact search stage for each depth.*

Proof. The construction and stabilization times give the first assertion. If $m(n)$ existed, evaluate $h_{m(e+1)}(e)$ to compute $H(e)$. The same computation works given any computable $s(e+1)$ with an exact stage. Exactness of the stage even just up to unpointed equivalence gives the same conclusion by Lemma 10.3. This contradicts the undecidability of the halting set [31, 6]. □

There is no computable bound dominating all s_n. This includes any proposed total computable bound based on the number of agents, the maximum bit index, and a fixed computable rule for dividing the work. Parallelism can change the cost of a finite simulation and can improve throughput. A computable scheduler with finitely specified controllers still generates a computable simulation. A service with an oracle, or a schedule supplying answers as non-computable external data, changes the hypothesis.

A positive result $H(e) = 1$ can be certified by a finite halting trace. If negative results also had finite certificates that a total computable verifier could check soundly and that existed for every negative case, the two searches could be dovetailed. On input e run the simulated program and enumerate proposed negative certificates in parallel. One search eventually succeeds and the two successes are incompatible. The resulting decider would contradict the same theorem. Restricted classes of programs can admit such certificates for all instances in the class. That is a different representation of the negotiated observation.

$$\begin{array}{ccc} X_{n+1}^{h_s} & \xrightarrow{\text{restrict}} & X_n^{h_s} \\ {\scriptstyle \text{revise}}\downarrow & & \downarrow{\scriptstyle \text{revise}} \\ X_{n+1}^{h_{s'}} & \xrightarrow[\text{restrict}]{} & X_n^{h_{s'}} \end{array}$$

Figure 2: At fixed search stage the horizontal maps restrict depth. In the split halting family the vertical maps add factors by the basepoint inclusions. At fixed depth the vertical maps become identities after stabilization.

12.3 A characterization by limit computability

For the bit family the construction generalizes to any total computable approximation $a : \mathbb{N} \times \mathbb{N} \to \{0, 1\}$ with pointwise eventual value $b(e)$. The corresponding finite word at stage s is

$$a_s|_n = (a(s, 0), \ldots, a(s, n - 1)).$$

At any fixed n finitely many coordinates are involved, so taking the maximum of their stabilization times gives eventual agreement of the whole word. Unlike in the halting example, a coordinate can change several times. The horizontal tower of a single row still has projection maps.

Theorem 12.2 (Characterization of tracked finite access)**.** *For the bit family the following are equivalent.*

1. *b is limit computable, meaning that there is a total computable $a(s, e)$ with eventual value $b(e)$ for each e.*
2. *There is a total computable generator of finite-stage Kan codes $Y_{n,s}$ such that at every fixed n the space $Y_{n,s}$ is eventually equivalent to X^b_n.*
3. *There is a generator as in condition 2, with a uniform coherent tower of hypotheses for each s, built from finite words and restriction.*

A computable modulus of eventual correctness in condition 2 exists for some such generator exactly when b is computable.

Proof. From condition 1 use the finite words $a_s|_n$ and their projection maps. Eventual agreement follows from finite stabilization, giving condition 3. Condition 3 entails condition 2. Under condition 2, given s, e, inspect degree $e + 1$ of the pointed homotopy invariant of $Y_{e+1,s}$ by Lemma 10.3. Choose a vertex if one is available and output 1 when the group order is 2, and 0 otherwise, also using 0 for an empty complex. The full list in degree zero tells which case applies. At sufficiently large s the complex is connected and equivalent to X^b_{e+1}, so the bit is $b(e)$. This gives condition 1. If the stage generator has a computable modulus, apply the recovery calculation at that modulus to compute b. Conversely, for computable b the exact uniform stage generator can be used at every s with modulus zero. □

Classically, by the Shoenfield limit lemma, limit-computable sets are exactly the sets computable in the halting oracle, or equivalently the Δ^0_2 sets [6, 4]. The two directions can be read directly here. If $a(s, e)$ stabilizes, an oracle for halting can answer whether the effective search for a later change from $a(s, e)$ terminates.

Search for a stage at which it answers no. Conversely simulate any given oracle computation with a finite approximation to the halting oracle, stopping after s steps and using a default value if it has not completed. A correct terminating run uses only finitely many oracle queries. Their answers eventually stabilize, and the simulated run at a sufficiently large stage follows the correct computation. This proves the two reductions and explains their relevance to a resource budget.

A pointwise approximation can be seen as convergence in the usual ultrametric on bit sequences:

$$d(b,c) = 0 \text{ if } b = c, \qquad d(b,c) = 2^{-\min\{e : b(e) \neq c(e)\}} \text{ otherwise.}$$

Agreement on the first n bits means distance at most 2^{-n}. For a computable sequence of hypotheses, a computable modulus at these dyadic radii would give the exact stage generator by reading the stable prefix. The metric describes the resolution of an observer. It does not give a computable rate of convergence.

Theorem 12.3 (Tracked access to extension classes)**.** *Let Y^b be the family with fixed graded homotopy groups from Section 11. The equivalences of Theorem 12.2 hold with X^b_n replaced by $\tau_{\leq n} Y^b$. The least depth that determines the e-th bit from the homotopy type of a stage, uniformly over all bit sequences, is $d_e = 2m_e - 1$.*

Proof. At depth n, a presentation of the named stage requires only the bits of the full blocks with $d_e \leq n$. Each such list is finite and the visible indices are computable from n. Lower layers of blocks whose upper groups are beyond n have a fixed description independent of their bits. Thus each row a_s of a limit-computable approximation gives a uniform tower of hypotheses using the canonical projections and block truncation maps. At any fixed depth, take a stage past the last change among the finitely many relevant bits. From then on the stage has the exact description. A uniform coherent family in particular gives stage hypotheses without a coherence condition.

Conversely, let $Z_{n,s}$ be any generator of eventually correct stage codes. On input (s, e) run the finite lift diagnostic of Proposition 11.3 against $Z_{d_e,s}$, using the first vertex if there is one. Output 0 if the diagnostic succeeds, 1 if it fails, and any fixed bit for an empty code. The algorithm only requires a degreewise finite Kan code, so this is total even for a hypothesis with the wrong homotopy type. For sufficiently large s the target is equivalent to $\tau_{\leq d_e} Y^b$, where the proof of Theorem 11.4 identifies the output with $b(e)$. A computable choice of a correct stage for each n therefore computes b. If b is computable the exact codes can be returned at every s, giving modulus zero.

The same lift diagnostic proves sufficiency of depth d_e with an exact input. If two sequences differ only at e, then every stage below d_e has the same canonical model for both sequences. Thus no rule that depends only on the homotopy

type of such a stage distinguishes their e-th bits. This proves the lower bound in observation depth. □

12.4 Changing a budget is not restriction in depth

Suppose $M_e(e)$ halts for the first time between s and s'. At fixed $n > e$, the two stage hypotheses differ in π_{e+1}. A natural transformation of towers can sometimes be supplied, for example inserting a zero cocycle in a new group factor for the monotone halting approximation. Such a comparison is not an equivalence at that depth. A diagonal selection $n \mapsto X_n^{h_{s(n)}}$ has no guarantee that the canonical projection of the selected stage at $n+1$ is the selected one at n. For an explicit instance choose a program index e that takes more than one step to halt, choose $n > e$, and let $s(n)$ be before that halt and $s(n+1)$ after it. The projection at depth n of the later word has bit 1 at e, whereas the earlier selected word has bit 0. Since the spaces have different group orders, an equivalence cannot repair the projection. In that sense increasing the search budget is revision, while moving down a coherent tower is restriction.

If a negotiation uses several learned tower objects, their pointwise stabilization gives eventual correctness of the composite at any fixed finite collection of depths for which all legs are eventually correct as diagrams. Stagewise equivalence of apices alone does not suffice when the maps into the mediator change. One must stabilize the data of the cospan up to equivalence of cospans. For the canonical bit tower and the constant basepoint legs, stabilization of the finite word supplies that stronger condition.

12.5 Resources as a cofinal access condition

Let B range over a preordered collection of resource entitlements. Let $C(n, s)$ be a sufficient requirement to run the hypothesis generator for depth n and stage s, including the entitlement to use its verifier and to store its returned code. An implementation can measure or overestimate this requirement for bounded simulation. Assume there is an increasing family of requirements $C^*(n, s)$ dominating $C(n, i)$ for all $i \leq s$. In the vector model one may use coordinatewise maxima for alternatives with reusable capacity and sums for computations that are actually run in sequence.

Definition 12.4 (Cofinal search service)**.** A service available to an agent is cofinal at depth n if, for every fixed s, it eventually allows a completed hypothesis computation at some stage at least s, and its completed stage indices tend to infinity along the observations under discussion. Availability includes required rights and the time or scheduling guarantee needed to complete the computation.

Corollary 12.5 (Resource-enabled observation)**.** *For a limit-computable bit object, every agent with a cofinal search service at depth n eventually obtains exact n-stage observations. For the halting object there is no general computable rule assigning a sufficient search stage as a function of depth. Cofinality of a service does not provide such a rule.*

Proof. Take a stabilization time s_n for the chosen learner. Eventually all completed indices are at least s_n, so the observations are exact. A general computable rule would provide a computable choice of exact stage, contrary to Theorem 12.1. □

The quantifier order in this result is $\forall n\, \exists s_n\, \forall s \geq s_n$. The existence of s_n is sufficient for a mathematical resource threshold at each fixed depth. Requiring a computable function of n is a stronger demand. An agent can also buy an exact finite word from a counterparty. Uniform dependable purchase over all depths then uses a source of non-computable information when b is non-computable. The code of a finite receipt can still be entirely effective once the word is supplied.

12.6 Exact queries determined by finite information

A Boolean observable depending only on the first n homotopy group orders has an eventual value for any learner in Theorem 12.2. Compute the orders on its stage hypothesis and evaluate the Boolean rule. When the word is correct, so is the answer. If the observable is instead the existence of a particular matching witness, one may first need one additional depth on the mediator by Theorem 8.1. For example, when $\mathbf{1} \to X^H \leftarrow \mathbf{1}$ uses the basepoint, the observer of ΩX^H in degree $j \geq 1$ reads the bit $H(j)$. Detecting that bit uses depth $j + 1$ of the mediator. At depth zero the group A_0^H is visible as distinct components of the match. This supplies a negotiation interface in which a resource acquisition affecting the allowable depth changes an observable of the actual composite.

13 Revisable observations and safe commitments

13.1 Positive evidence in a changing display

The halting approximation has extra structure because $h_s(e) \leq h_{s'}(e)$ when $s \leq s'$. At a coordinate that changes, the map between factors of the split family is the basepoint inclusion $\mathbf{1} \to K(\mathbf{C}_2, e + 1)$. At an unchanged coordinate it is the identity. Taking products gives maps $X_n^{h_s} \to X_n^{h_{s'}}$ commuting with the horizontal projections. There are also compatible maps to X_n^H. The inclusions and their retractions by forgetting the new factors are computed uniformly from the finite hypotheses when both search indices are given.

This structure supplies a direct notion of sound evidence before stabilization. Given a functorial witness type $W_{n,s}$ of a property under these maps, a point of $W_{n,s}$ can be transported to the ideal witness type if the property is covariant and the target comparison is part of the specification. Existence is preserved under any such map. An assertion that a property has no witness is contravariant in general. It cannot be inferred at the ideal object from emptiness at a stage. When the comparison has a retraction on the relevant witness types, equality or inequality of the images of earlier witnesses can also be analyzed by applying that retraction.

Proposition 13.1 (Sound stage certificates)**.** *Fix a physical boundary $U \times V$ and suppose that a stagewise witness span $U \leftarrow W_s \rightarrow V$ has a specified map over its boundary to an ideal span $U \leftarrow W_\infty \rightarrow V$. For a query specified by a probe $T \rightarrow U \times V$, any map $T \rightarrow W_s$ over the boundary yields a map $T \rightarrow W_\infty$ over the boundary. If the first map is available as an effective witness, an effective controller may validate it under a rule expressly accepting that stage certificate. If the comparison has a retraction over the boundary, distinct components of the stage mapping space remain distinct in the ideal mapping space.*

Proof. Compose the stage witness with the comparison, using its boundary paths to supply the over-map. The meaning of validation in the second assertion is that the stored rule checks the stage input and its authority conditions. It need not compute the ideal map. For the third assertion apply the induced map on mapping spaces to the retraction over the boundary. The composite of the comparison and the retraction is homotopic to the identity. Taking π_0 therefore gives an injective map on components. □

The retraction clause explains when a nontrivial stage loop remains nontrivial. It cannot be replaced by arbitrary monotonicity of stages. A map $BG \rightarrow \mathbf{1}$, for example, sends all group loops to the identity. If the acceptance criterion is that a supplied loop remains non-null in the ideal contract, a controller needs a condition excluding this collapse. In the split halting family the factor inclusion and projection give exactly that condition, and a finite halting trace can certify that the corresponding coordinate really occurs.

13.2 A variance test for acceptance conditions

Consider three kinds of Boolean questions an agent may ask about $X_n^{h_s}$. First, the assertion that the distinguished slot e has an activated $\mathbf{C}_2$ factor is positive in the monotone learner, and a positive answer has a finite halting trace. Second, the assertion that all slots below n are trivial can change from true to false. A trace showing no halt in a bounded interval does not make this second answer

stable. Third, the assertion that exactly one slot is active can change in either direction along a growing search, from false to true and then to false. These three cases are separated by the order on bit prefixes, not by whether their evaluation on a finite word is effective.

For a monotone Boolean rule $F : \{0,1\}^n \to \{0,1\}$, positive answers on $h_s|_n$ are sound for $H|_n$. They are also complete in the limit: if $F(H|_n) = 1$, then all hypotheses after s_n return one. Monotonicity proves soundness, and prefix stabilization proves the eventual completeness. An antitone rule preserves a negative verdict under growth in s, with the same eventual completeness for negative verdicts. An arbitrary Boolean rule need only stabilize. This distinction gives a simple algebraic method to choose whether a contract can safely authorize an irreversible action on a provisional report.

The higher lift test of Section 11 provides an important contrast. Its answer yes corresponds to the bit being zero. Under monotone learning for H, yes can be retracted when a twisted block becomes visible. The fact that it is a positive existential statement about a *map* does not make it monotone in the learner. A split block cannot have a comparison to the twisted block that preserves the detecting π_m class: it would carry the split lift witness to a detecting lift in the twisted block, contradicting Proposition 11.3. This shows why a variance claim must use the actual transitions of the witness family rather than the logical syntax of the query in isolation.

13.3 Settlement by versions

A protocol may accept a bounded observation at version s, while promising to issue newer versions as resources permit. It can attach the search stage, depth, and exact operational claim to the signed record. A future verifier can then distinguish a correct bounded claim from a statement about the limit. The ledger can allow a new contract to use the later evidence without treating the earlier contract as if it had received that evidence. A release or compensation is a new authorized transition, with resources and recipients specified in its own rule.

Suppose a query is sound in the sense of Proposition 13.1 and its certificate persists under future search versions. A controller can authorize the corresponding settlement after an independent ledger check. Persistence ensures that re-evaluating the semantic claim at a later version does not invalidate the reason for that authorization. If the contract instead requires a property of the earliest version at which a witness exists, then preservation of witness existence is insufficient to validate the claimed earliest time. That minimum is a statement about the entire prefix of the computation. A verifier would need a history proof of the negative observations up to the reported first version, interpreted at their

specified stages.

13.4 The order of two infinite operations

Let $A_{n,s}$ be a diagram with restrictions in n and comparison maps in s. There is a canonical map

$$\operatorname*{colim}_{s} \lim_{n} A_{n,s} \longrightarrow \lim_{n} \operatorname*{colim}_{s} A_{n,s}.$$

On the left one first realizes each coherent row and then combines those realizations. On the right one allows each finite observation to stabilize before realizing the tower. These operations do not commute in general, even for diagrams of finite sets. Take $A_{n,s}$ to be the binary words of length n with all entries at indices $e \geq s$ equal to zero. Use prefix restriction in n and inclusion in s. For fixed s the limit is the set of infinite binary words supported below s. Its filtered union is the set of finitely supported words. For fixed n, the colimit is all words of length n, and the inverse limit is all infinite words. The comparison is inclusion of the finitely supported words and is not surjective. All sets are used as discrete spaces, so these calculations are also calculations of the relevant homotopy limits and filtered colimits.

For the split Eilenberg–Mac Lane bit family with the monotone halting approximation, the comparison is an equivalence. This is a special feature of the degreewise bounded dependence. At a fixed simplicial degree j the infinite product and every row of the tower depend only on the finitely many bits with $e + 1 \leq j$. Those bits have stabilized for all $s \geq s_j$. In simplicial sets, filtered colimits of Kan complexes are Kan and compute the filtered colimit in spaces. To see the Kan condition, a horn is finite data and therefore comes from some finite stage of the filtered diagram, where it has a filler. Homotopy groups are computed by finite spheres and homotopies, so the same filtered comparison computes the homotopy type. The row limits are represented by the product-projection fibrations, and the column colimits stabilize at each named n. Consequently both sides have the canonical Kan model of X^H. The proof establishes equivalence at the level of spaces. It does not give a computable stage at which a single row realizes the limit.

13.5 Finite distributed decisions

A network can attach a rule to each settlement named by a finite collection of depths and probes. If all arguments of the rule are computable queries on the finite hypotheses, its verdict at a given stage is computable. If the hypotheses stabilize on the finite collection, and the verdict depends only on the homotopy-invariant queries, the verdict eventually becomes correct. This follows by taking

the maximum of the finitely many stabilization stages and applying the stipulated maps on witnesses. The resulting rule can support a revisable discovery service.

For a commitment service, classify the verdicts by which ones have a sound finite certificate. A participant can then distinguish a verified action from an unverified recommendation without learning a computable stabilization modulus. In the halting family the distinction is forced by Theorem 12.1. If a network gave a uniformly correct definitive bit for every input merely by waiting until a computed threshold, simulating that network would decide the halting set. The positive fragment avoids that contradiction by keeping absence of an accepted witness as an operational state rather than a certificate of a negative answer.

14 Pro-objects, realization, and the problem of choice

14.1 A pro-object represents finite interaction

Let $\mathrm{Pro}(\mathcal{F})$ be the pro-category of finite observation spaces, with size taken in a larger universe if needed. A cofiltered diagram $X : I \to \mathcal{F}$ represents a pro-object. For two such diagrams the mapping space is

$$\mathrm{Map}_{\mathrm{Pro}(\mathcal{F})}(X, Y) \simeq \lim_{j \in J} \operatorname*{colim}_{i \in I^{\mathrm{op}}} \mathrm{Map}_{\mathcal{F}}(X_i, Y_j) \tag{12}$$

with the usual contravariance in the source [1, 27]. For a constant target Z this is the filtered colimit of maps from finite stages. A connected component of this mapping space has a representative at a sufficiently deep stage. A higher path between two represented maps can also be compared after passing to a common deeper stage, because finite simplicial data commute with filtered colimits in spaces.

The functor taking a named tower to its pro-object permits a cofinal change of the stage index. A protocol that charges according to exact observation depth can use the named tower before this passage. A protocol that only asks whether every finite target query factors at some finite stage can use the pro-object. Choosing one semantics affects which distinctions are preserved. Neither construction attaches a time cost to a factorization unless the presentation and a cost rule supply it.

Proposition 14.1 (Products and finite seams in the pro-category)**.** *A finite cospan of towers on one cofiltered index category, with specified natural legs, has its pro-pullback represented by the pointwise homotopy pullback tower. The same holds for the terminal object and for finite products of such diagrams.*

Proof. For a pro-test object T, apply (12) to the pointwise pullback tower. At each target index j, the map from a source stage T_i into a pullback is the pullback of

the three mapping spaces. The filtered colimit over source stages commutes with finite limits in spaces. The limit over j commutes with the same finite limit. Hence the resulting mapping space is the homotopy pullback of the three pro mapping spaces from T. This is the pro-category universal property. The argument with products or a terminal diagram is the same finite-limit interchange. □

The proposition starts with a common index, as happens with the resource observation levels of the paper. General maps between pro-objects need not come already in that form. The finite observation statement requires only the chosen displayed cospan. For a family of objectives that can query a constant target, (12) says each query factors at some stage. It says nothing about a single uniform depth at which all objectives factor.

14.2 Realization preserves limits and can add maps

View a pro-object in $\mathrm{Pro}(\mathcal{S})$ and define its realization to be the limit of its diagram in $\mathcal{S}$. Mapping out of a constant space T gives

$$\mathrm{Map}_{\mathrm{Pro}(\mathcal{S})}(cT, X) \simeq \mathrm{Map}_{\mathcal{S}}(T, \mathrm{holim}\, X).$$

Thus realization is right adjoint to the constant-space inclusion, in suitable universes. It is independent of a cofinal replacement and preserves limits. Restricting to $\mathrm{Pro}(\mathcal{F})$ still gives a well-defined realization in spaces. This functor need not be full. A map out of the realized space can depend on all coordinates at once in a way no finite stage records.

Example 14.2 (A map with no finite representative)**.** Let $G_n = (\mathbf{C}_2)^n$, with coordinate projections, and $P_n = BG_n$. The maps $P_{n+1} \to P_n$ are Kan fibrations, as can be seen from the surjectivity of the group homomorphisms and the horn condition for the nerve of a groupoid. The strict inverse limit is BG for the abstract group $G = \prod_{e\in\mathbb{N}} \mathbf{C}_2$, since taking the nerve is levelwise finite powers and commutes with limits. Thus the realized space is the classifying space of that abstract group.

For a constant target $B\mathbf{C}_2$, the connected components of the pro mapping space are the filtered colimit of the homomorphisms $G_n \to \mathbf{C}_2$, with trivial conjugation in the target. Their realizations are exactly the abstract homomorphisms $G \to \mathbf{C}_2$ which factor through one of the coordinate projections. Such a homomorphism is continuous for the product topology with discrete target. Conversely continuity forces its kernel to contain a basic open subgroup, so it factors through finitely many coordinates.

By classical linear algebra there is a nonzero linear functional on

$$G/\bigoplus_{e\in\mathbb{N}} \mathbf{C}_2$$

over the field with two elements. For example, the constant sequence of ones has nonzero class. Extend it to a basis and send this class to one and the other basis elements to zero. The resulting homomorphism $G \to \mathbf{C}_2$ vanishes on the direct sum. A finite-stage homomorphism vanishing on that dense subgroup is zero, since the subgroup surjects onto every G_n. The nonzero homomorphism gives a map $BG \to B\mathbf{C}_2$ with no pro representative. This establishes the claimed difference without assuming that finite targets always force finite dependence after realization.

The example shows why the ideal semantics of all maps between realized objects is broader than finite negotiation. If a provider asserts the existence of a map in the ideal space, finite observability is a separate requirement. A morphism to a finite target in the pro-category satisfies a finite factorization theorem. A morphism merely in spaces need not do so. The bit family is compatible with this distinction. Its named Postnikov stages can be presented separately, the pro-object permits cofinal changes, and the homotopy groups of its realization still recover the bit sequence.

14.3 An effective tower can lack an effective coherent choice

Theorem 14.3 (Coherent-choice separation)**.** *There is a uniform effective tower of nonempty finite discrete spaces with an inhabited homotopy limit in classical mathematics and no computable coherent point.*

Proof. Use an effective enumeration (φ_e) of partial computable functions. Set

$$U = \{e : \varphi_e(e) \downarrow = 0\}, \qquad V = \{e : \varphi_e(e) \downarrow = 1\}.$$

The sets are disjoint and computably enumerable. For n let T_n consist of binary words of length n such that for $e < n$, if simulation of $\varphi_e(e)$ for n steps returns 0 then the e-th bit is 1, and if it returns 1 then the bit is 0. Each T_n is a finite set with a uniform total listing. It is nonempty because the constraints never demand both values at one coordinate. Restriction of a word of length $n+1$ to length n remains valid, since every constraint seen at stage n is also seen at stage $n+1$. This gives the tower.

The limit is the set of binary sequences separating U, V in the indicated directions. Such sequences exist, for instance the characteristic function of U. If a limit sequence were computable, choose an index e for that total binary function in the enumeration. If its value at e is 0, the index lies in U, requiring value 1. If its value is 1, the index lies in V, requiring value 0. This contradiction proves there is no computable coherent point. Discreteness identifies the ordinary compatible choice with the homotopy-limit point. □

The theorem concerns points of a particular presented tower. The assertion that the tower has an effective code is not a claim that every, or any, global point of its limit has an effective code compatible with those projections. A controller can produce a valid word at each requested depth, for example the first in its finite listing. Those independently chosen words need not extend one another. Requiring irreversible commitment at every stage with no revision is the extra difficulty in a negotiation interpretation.

14.4 Where finite connectedness helps

There is a useful exact statement about components of towers with finite observation spaces.

Proposition 14.4 (Components of finite-stage limits)**.** *For a sequential tower A of π-finite spaces, the map $\pi_0 L(A) \to \lim_n \pi_0 A_n$ is a bijection in classical mathematics with countable choice.*

Proof. For surjectivity choose one point in each specified compatible component and a path witnessing compatibility of the chosen points. These choices give an element of $L(A)$. For injectivity suppose $(x, \xi), (y, \eta)$ have the same component at each stage. At stage n let Q_n be the set of components of the path type $x_n = y_n$. This set is nonempty and finite, since a chosen path identifies that type with a loop space of a π-finite space. There is a transition on Q_n induced by

$$q_{n+1} \longmapsto \xi_n^{-1} \cdot \mathrm{ap}(r_n, q_{n+1}) \cdot \eta_n.$$

The finitely branching tree argument gives a compatible sequence of path classes. Choose representative paths q_n in those classes. Compatibility of classes gives, for every n, a higher path expressing the equality in (7). Choose those higher paths. Then Lemma 7.1 gives a path between the two limit points. □

Finiteness in this argument occurs in the path classes, not solely in the components of the original spaces. For comparison, take the tower $B\mathbb{Z}$ with the maps induced by multiplication by 2. Every stage has one component. A compatible family can be represented, after choosing the designated vertices, by a sequence of seam loops $g_n \in \mathbb{Z}$. By (7), two such sequences are in the same component exactly when

$$g'_n - g_n = a_n - 2a_{n+1} \quad \text{for some sequence } (a_n) \in \mathbb{Z}^{\mathbb{N}}.$$

There are no further identifications of the integers inside the loop space of $B\mathbb{Z}$. Thus the set of components is the quotient by the indicated difference operator. The map $g \mapsto \sum_{n \geq 0} 2^n g_n$ converges in $\mathbb{Z}_2$. It sends the image of the difference

operator to the subgroup $\mathbb{Z}$, by telescoping. If the sum is an ordinary integer a_0, set

$$a_n = \frac{a_0 - \sum_{k<n} 2^k g_k}{2^n}.$$

The congruence defining the 2-adic sum makes each a_n an integer, and $a_n - 2a_{n+1} = g_n$. Every 2-adic integer has a binary expansion, giving surjectivity of the sum map. Consequently the component set is $\mathbb{Z}_2/\mathbb{Z}$, while the inverse limit of the stage component sets is a singleton. This is an explicit instance of the derived-limit phenomenon for towers [3]. It illustrates why the path data in a coherent agreement matter even when feasibility is recorded just as an inhabited component.

15 Composing canonical observation towers

15.1 Levelwise gluing and the actual Postnikov tower

Let $P = A \times^h_B C$, and consider the two towers

$$P_n = \tau_{\leq n} P, \qquad Q_n = (\tau_{\leq n} A) \times^h_{\tau_{\leq n} B} (\tau_{\leq n} C).$$

Both consist of n-types at depth n. For Q_n this follows from closure of n-types under limits. It also follows synthetically from the sum of path fibers description: the base is a product of n-types and the path fibers are $(n-1)$-types. The natural cone $P \to Q_n$ factors through P_n, giving a comparison $a_n : P_n \to Q_n$. By Proposition 8.2, this comparison can fail to be an equivalence at the named depth. The relevant loss arises before passing to an inverse limit.

There is also a comparison in the reverse direction from one depth higher. By Theorem 8.1 the map

$$d_n : P_n \longrightarrow \tau_{\leq n} Q_{n+1}$$

induced by the same cone is an equivalence. Define $b_n : Q_{n+1} \to P_n$ as the truncation unit followed by d_n^{-1}. These comparisons form an interleaving of the two towers. The interleaving is a precise way to express the dimension price of matching. It gives a uniform conversion between requests at different observation depths.

Theorem 15.1 (One-step comparison of towers)**.** *The maps $a_n : P_n \to Q_n$ and $b_n : Q_{n+1} \to P_n$ can be taken coherently in n, and their composites $a_n b_n$ and $b_n a_{n+1}$ agree with the respective restrictions $Q_{n+1} \to Q_n$ and $P_{n+1} \to P_n$. They give an equivalence of the associated pro-objects. When A, B, C have stages in $\mathcal{F}$, both towers can be regarded as pro-objects of $\mathcal{F}$.*

Proof. The maps a_n are obtained by the naturality of truncation and the universal property of P_n. In particular they commute with the restriction maps, with the coherent comparisons inherited from the natural cone. The maps d_n assemble into a natural transformation from $P_\bullet$ to the tower whose term at n is $\tau_{\leq n}Q_{n+1}$. The transition in the latter tower is induced by restriction of Q followed by truncation at the lower degree. Since d_n is an equivalence at every n, this is an equivalence of diagrams. Choose its inverse as an equivalence of diagrams. The units $Q_{n+1} \to \tau_{\leq n}Q_{n+1}$ are natural for these transitions. Their composite with that inverse gives the coherent b_n.

For $b_n a_{n+1}$, precompose with the unit from P to P_{n+1}. The result agrees with the unit to P_n by the definition of d_n. The universal property of P_{n+1} for maps to the n-type P_n identifies the composite with restriction. For $a_n b_n$, factor both it and the restriction $Q_{n+1} \to Q_n$ through $\tau_{\leq n}Q_{n+1}$. This is possible because Q_n is an n-type. On precomposition with d_n, the two factored maps agree with a_n, by the naturality of the cone from P. Since d_n is an equivalence, the factored maps agree. The comparisons follow from naturality and the universal properties, giving their coherent versions in the diagram category. Passing to pro-objects, shifting the index by one is cofinal, and the two restrictions become the identity comparisons for the represented pro-objects. Thus the two maps are inverse there.

When the input stages are in $\mathcal{F}$, the Q_n are in $\mathcal{F}$ by Lemma 4.3. The P_n are equivalent to $\tau_{\leq n}Q_{n+1}$. For any degreewise finite Kan S, the $(n+1)$-coskeleton is a degreewise finite Kan model of $\tau_{\leq n}S$: the same argument as in Lemma 4.3 preserves all homotopy through degree n and kills the higher groups. Thus P_n also lies in $\mathcal{F}$. □

The distinction between P_n and Q_n is relevant to a fixed-depth contract. The equivalence of pro-objects gives a correct finite observation after allowing the conversion of depth. It does not identify the two native depth-n answers. An agent with a budget for depth n in all the inputs may need one additional level of resources to realize the comparison. After allowing cofinal changes, this finite cost in depth does not change the represented pro-object.

15.2 An effective implementation of the corrected tower

Suppose the input Postnikov towers have uniform strict presentations and the legs at all stages are computable maps of the towers. This is a stronger hypothesis than separate stage computability. The path-object algorithm gives a uniform strict presentation of $Q_\bullet$, by Theorem 4.5. To compute a presentation of P_n, take Q_{n+1} and form its $(n+1)$-coskeleton. The stage code takes the already computed complete lists through degree $n+1$ and uses the finite assignment procedure for each requested higher simplex. The algorithm does not compute a minimal

Postnikov decomposition or try to decide validity of an arbitrary presentation.

To obtain the transition from the model of P_{n+1} to the model of P_n, use the transition $Q_{n+2} \to Q_{n+1}$ through degree $n+1$. In those degrees the two coskeleton models agree with their original complexes. The coskeleton adjunction extends the map uniquely to the required target in higher degrees. This supplies uniform adjacent maps. The construction is a strict tower model for $\tau_{\le n} Q_{n+1}$ as n varies. The natural equivalence d_n identifies its ideal interpretation with the Postnikov tower of the composite. Thus named-depth correction is both mathematically valid and effective on the stated data.

A similar calculation applies to finite graph diagrams of the form (5). Products and the diagonal give a finite cospan, so one can combine all seams before making the final correction. The procedure needs one further input depth independently of the size of the graph, although the size of the stage presentation and the work of matching can grow substantially. For a family indexed by a probe of dimension d, the combined result of Section 9 gives the comparison with $d+1$ extra degrees. This is an upper bound on depth, not a bound on computation time or on a learner's stabilization time.

15.3 Canonical towers and maps in the ideal semantics

For a space X write $\mathcal{P}(X) = (\tau_{\le n} X)_n$, viewed as a pro-object of the category of truncated spaces. Take universes large enough for the pro-category. A fixed m-type T has

$$\mathrm{Map}_{\mathrm{Pro}}(\mathcal{P}(X), cT) \simeq \operatorname*{colim}_n \mathrm{Map}(\tau_{\le n} X, T) \simeq \mathrm{Map}(X, T).$$

For the second equivalence restrict to $n \ge m$, a cofinal subsystem, and use the universal property of truncation. Each term is naturally equivalent to the same mapping space. In particular an ideal map to a truncated target is already determined at some finite depth for this canonical tower. If the target has a code, this factorization still describes a semantic map. A uniform procedure to supply its code depends on the effective hypotheses about source and map.

Proposition 15.2 (Full faithfulness on Postnikov towers)**.** *In spaces, taking the canonical Postnikov pro-object is fully faithful. In particular, on spaces all of whose Postnikov stages belong to $\mathcal{F}$, their canonical towers in* $\mathrm{Pro}(\mathcal{F})$ *recover the full mapping spaces between the realized objects.*

Proof. Apply the mapping formula (12) with $T = \tau_{\le m} Y$ in each target level. The calculation for a constant m-type gives

$$\mathrm{Map}_{\mathrm{Pro}}(\mathcal{P}(X), \mathcal{P}(Y)) \simeq \lim_m \mathrm{Map}(X, \tau_{\le m} Y) \simeq \mathrm{Map}(X, Y).$$

The last step uses convergence of the Postnikov tower in spaces and preservation of limits by the representable mapping functor. The calculation respects composition because the comparisons are induced by the canonical maps from X, Y. The subcategory $\mathcal{F}$ is full, so the same mapping formula applies to its towers when all the required stages lie there. □

This proposition has a narrower domain than the comparison in Section 14. The tower $B((\mathbf{C}_2)^n)$ from Example 14.2 is not the Postnikov tower of its realization. It changes the fundamental group at arbitrarily many stages, while a canonical Postnikov tower would stabilize that group at degree one. Hence the existence of a map of realizations with no finite representative in that example is consistent with full faithfulness on canonical towers. An agent's limited observation can therefore have two distinct causes. It can come from a named homotopy degree within a canonical Postnikov tower, or from a stage that observes only part of the data in a fixed degree. The pro-mapping formula measures the latter without identifying it with truncation.

15.4 A change-of-context consequence

In the category of spaces, the class of objects with all Postnikov stages in $\mathcal{F}$ is closed under finite limits. For a cospan of such spaces, Theorem 15.1 constructs the finite stages of the pullback. The terminal space has constant finite stages. The canonical Postnikov pro-object assignment preserves these finite limits, by the equivalence with the levelwise limit and Proposition 14.1. A finite network of canonical tower objects can therefore be composed before or after forming its pro-observations, provided depth changes of the indicated kind are allowed.

For a context where the canonical tower is known to converge, the same argument for mapping into truncated targets is available if the relevant slice admits the truncations and their universal properties. In an arbitrary ∞-topos, the full faithfulness argument identifies the mapping space into the Postnikov limit of the target. Identifying that target with its limit is a separate hypothesis. The finite-observation laws through Theorem 8.1 depend on paths and truncation and can be used internally where their premises hold, while the conclusion about global realization must record the chosen topos.

Part IV
Negotiation and applications

16 Strategies, payoffs, and approximate negotiation

16.1 The strategic datum

A witness space describes what can be jointly realized. A strategy specifies choices at histories or observations. Let S_i be the strategy type of participant i for a fixed horizon, information structure, and rule for fuel. For perfect recall a pure strategy gives one action at every possible local history where that participant acts. In an effective system it can be a code computing those actions. A mixed strategy can be introduced by a probability distribution, which we use only when stated. The resulting play of a profile depends on the environment and scheduler. A payoff function is defined after those dependencies have been fixed or given a probability law.

A feasible profile has a joint witness in the network, with constraints on legal deviations. This can be a generalized game: whether a replacement s_i' is feasible may depend on the strategies of the other participants. For instance, a provider cannot unilaterally switch to a sale that requires the buyer to sign a different bundle. On the other hand, a unilateral choice to reject must be included when the controller actually has that right. The difference concerns the space of allowed deviations, which cannot be reconstructed from the set of signed contracts alone.

Write $F \subseteq \prod_i S_i$ for the allowed profiles in a set-valued presentation, and

$$D_i(s) = \{x_i : (x_i, s_{-i}) \in F\}$$

for its allowed unilateral replacements when legality is determined by F. More generally the protocol can specify a smaller or otherwise contextual deviation relation. Let $u_i : F \to \mathbb{R}$ be the payoff. We assume all payoff values and suprema appearing below are finite. The regret is

$$\mathrm{Reg}_i(s) = \sup_{x_i \in D_i(s)} \big(u_i(x_i, s_{-i}) - u_i(s)\big),$$

where the current strategy is included in $D_i(s)$. A δ-equilibrium satisfies $\mathrm{Reg}_i(s) \le \delta$ for every i. An outside option can be a designated payoff d_i, with individual rationality expressed as $u_i(s) \ge d_i$. This is distinct from equilibrium, which compares all admissible unilateral deviations.

In a higher presentation, strategy types and feasibility types can carry paths. A real-valued payoff invariant under witness equivalences factors through the

set of components of its domain by truncation elimination. The feasibility and deviation relations must also specify how their witnesses are transported. A higher equivalence that identifies the feasibility type can change strategy semantics if it discards which agent chooses which coordinate. For comparison of games we require transport on the strategy and deviation data as well as the payoff function.

Proposition 16.1 (Transport of a strategic problem)**.** *Suppose an equivalence between two presentations gives a bijection of feasible profile components and, at each profile, a bijection of the admissible deviation components for each agent, respecting the resulting profile and payoff. Then it preserves the regrets, the equilibrium components, and individual rationality with the same outside options.*

Proof. For each agent the possible payoff gains are the same set of real numbers, indexed by the stated bijection of deviations. Their suprema are equal. The payoff at the current profile is also equal, so comparison with the same outside option gives individual rationality in both presentations. □

This proposition states the extra data that a strategic equivalence needs. An equivalence of spans such as Proposition 6.1 preserves which matching witnesses exist. A game-theoretic construction can use that equivalence, but must still show that it preserves the underlying unilateral deviations. Open games provide a different composition law that incorporates strategic context directly [9, 10].

16.2 Feasibility does not imply strategic stability

Example 16.2 (Opposing preferences at one seam)**.** Two agents each choose a bit. Every pair has a feasible agreement witness, with a one-point mediator. The first agent receives 1 when the bits agree and 0 otherwise. The second receives 1 when they disagree and 0 otherwise. Every profile has a unilateral deviation increasing one payoff by one. There is no pure equilibrium. The feasibility witness calculus is exact, because the four joint choices are present. The failure concerns the dependence of a participant's payoff on the other participant's choice.

A second failure involves individually rational local trades. Suppose one provider must purchase a verifier service before delivering to a buyer. The buyer and provider each consent to their local witness given a fixed quoted verifier price. A unilateral deviation by the verifier may raise that price if the downstream buyer's surplus remains positive. The two local agreements can glue over the quoted price, while the composite is unstable in a game where the quote can be changed. An expiry or a binding reservation changes the set of deviations. The effect must be represented in the strategy model, not inferred from the glued receipt.

16.3 A finite horizon and a discounted continuation

Fix the policies of all agents other than i and fix an effective scheduler for a finite-horizon negotiation. Suppose the remaining state space has a finite branching tree from the initial state, the available actions are effectively listed, and per-step rewards and the terminal reward are rational. A best response for i can be computed by backward induction when it has the modeled state information. In a partial-information version the state for the recursion must encode its information set or sufficient statistic, and feasible actions must be constant on indistinguishable states. The calculation below is for a fully observed state.

Let $r(s,a)$ be the immediate reward and $F(s,a)$ the next state. For discount $0<\gamma<1$ let

$$(TV)(s)=\max_{a\in A(s)}\big(r(s,a)+\gamma V(F(s,a))\big),$$

where the finite action set is nonempty. A terminal state may have one absorbing action with reward zero. The cost of communication, computation, or expiration can be included as a negative immediate reward. The model of F is determined by the other policies and scheduler. If a step is probabilistic, the same proof works with an expectation of V when the model specifies a transition probability.

Proposition 16.3 (Discounted response)**.** *Assume $|r(s,a)|\le R$ uniformly. The operator T is a contraction in the supremum norm on bounded value functions with factor γ. Its fixed point is the value of an optimal response, and the error of the horizon-t value T^t0 is at most $R\gamma^t/(1-\gamma)$.*

Proof. For any two bounded functions V,W, each candidate inside the maximum changes by at most $\gamma\|V-W\|_\infty$. The maxima therefore differ by at most this bound. Completeness of bounded functions in the supremum norm gives a unique fixed point. The fixed point has norm at most $R/(1-\gamma)$, because the bound $R+\gamma\|V\|_\infty$ applies to TV. Applying the contraction t times gives the stated error. At every state a maximizing action exists by finiteness. Following a maximizing action at each step iterates the fixed-point equality. The bounded tail tends to zero, proving that this policy realizes the fixed-point value. For any other policy iterate the fixed-point inequality to obtain an upper bound by the same value. Thus the fixed point is optimal. □

For the effective finite-horizon problem the recursion explores only finitely many branches at each finite horizon. Its use can still be uneconomic if the tree is large or the controller has little fuel. The theorem gives a uniform error bound as a function of the horizon when R,γ are known. This source of approximation has a computable modulus in the rational model, in contrast to the halting-based

homotopy observation of Section 12. Both can occur in one negotiation, and the stronger computational barrier can dominate a promise of exactness.

16.4 Stability under payoff error

Let $u_i, \widehat{u}_i$ be two payoff families on the same feasible profiles, with the same deviations. Suppose that

$$\sup_{s \in F} |u_i(s) - \widehat{u}_i(s)| \le \varepsilon_i. \tag{13}$$

Theorem 16.4 (Regret under a common feasibility model)**.** *At every profile s,*

$$\left| \mathrm{Reg}_i^{u}(s) - \mathrm{Reg}_i^{\widehat{u}}(s) \right| \le 2\varepsilon_i.$$

In particular, a δ-equilibrium for the approximate payoffs has regret at most $\delta + 2\varepsilon_i$ for agent i under the exact payoffs.

Proof. The difference between the exact and approximate gain of any admissible deviation is at most the error at the deviating profile plus the error at the baseline profile. Each is at most ε_i. Taking suprema over the same deviation set in both models gives the two inequalities and hence the absolute bound. □

The coefficient 2 is sharp. In a one-agent problem with two actions, take both approximate payoffs zero. Give the baseline exact payoff $-\varepsilon$ and the other action exact payoff ε. The approximate baseline has zero regret and the exact baseline has regret 2ε. The example does not use any choice of witness for the actions, so the bound also applies when those choices are represented by higher types.

Individual rationality has a one-sided error bound. If a profile has approximate surplus at least ε_i above its exact outside option, (13) implies that the exact surplus is nonnegative. If the outside option is itself approximated within error η_i, a margin of $\varepsilon_i + \eta_i$ suffices. The proof subtracts the two value estimates. A risk-averse verifier can reserve such a margin before recommending acceptance.

16.5 Changing what an agent can do

An observation can omit feasible deviations or introduce a provisional action that turns out to be illegal. For the following theorem we require that every approximate profile under comparison has a specified valid realization $\iota(\widehat{s})$ in the exact game. Let $s = \iota(\widehat{s})$. For each approximate deviation used in the argument, its outcome under ι is an exact unilateral deviation outcome from s. Suppose payoffs differ by at most ε_i on this baseline and all the approximate deviating

profiles. A coverage bound η_i means that for every exact deviation outcome s' from s, there is an approximate unilateral deviation outcome $\widehat{s}'$ from $\widehat{s}$ with

$$u_i(s') \le u_i(\iota(\widehat{s}')) + \eta_i.$$

This condition compares achievable payoff improvements. It does not require a bijection of action names.

Theorem 16.5 (Regret with incomplete observation)**.** *Under these realization, payoff, and coverage assumptions, if $\widehat{s}$ has approximate regret at most δ for agent i, then s has exact regret at most $\delta + 2\varepsilon_i + \eta_i$.*

Proof. Fix an exact deviation outcome s', and choose a covered approximate deviation $\widehat{s}'$. Then

$$\begin{aligned} u_i(s') - u_i(s) &\le u_i(\iota(\widehat{s}')) - u_i(\iota(\widehat{s})) + \eta_i \\ &\le \widehat{u}_i(\widehat{s}') - \widehat{u}_i(\widehat{s}) + 2\varepsilon_i + \eta_i \\ &\le \delta + 2\varepsilon_i + \eta_i. \end{aligned}$$

Take the supremum over exact deviations. Each bound concerns only baseline and deviating outcomes related by ι, so it applies also to contextual deviation sets. □

An approximate optimum can be arbitrarily bad without coverage. Let a one-agent observation expose only the action with payoff zero. In the exact problem add a legal action with payoff $M > 0$. The payoff error on all exposed actions is zero and the approximate regret is zero, but the exact regret is M. The same obstruction occurs if a finite observation misses a coherent delegation that makes another provider usable. Conversely a hypothetical extra action can make an approximate profile infeasible. In that case the realization hypothesis must be checked before applying the theorem.

16.6 Payoffs that respect observation depth

A value function on the ideal agreement space can be exactly determined at one finite depth if its map factors through that finite observation. For the bit family, one may instead give a uniform estimate. Let f be L-Lipschitz on the ultrametric space of bit words completed to infinite sequences. If an exact depth-n prefix is known and any completion is used, the estimate differs from $f(b)$ by at most $L2^{-n}$. This follows from the definition of the metric in Section 12. The bound assumes exactness of the prefix. An eventually correct learner produces the same bound after its stabilization time for that n, which can be non-computable in n.

If a search controller has a finite horizon with error from Proposition 16.3 and a payoff observation with known uniform error, one can add the two errors by the triangle inequality. There is no such addition for an unverified assertion that the horizon and observation describe the same feasible set. The coverage term in Theorem 16.5 is one way to make that compatibility explicit. Thus the theory distinguishes improving the accuracy of values from discovering genuinely new actions, a distinction that governs when acquiring more verification capacity pays off.

17 Acquiring the means to observe

17.1 A two-step purchase

Consider an agent with an initial resource entitlement B. It can buy a verification slot v by spending a transferable amount p. Its new entitlement is obtained by a guarded debit and credit. The slot allows it to evaluate a family of requests up to depth n. If the resulting information qualifies it to bid on a downstream compute contract, settlement of that contract can increase the available search budget for later requests. The coupling goes through an actual history. Observation depth is a property of the evaluator's available service, while the balance is updated by verified transitions.

Let $E(B) \subseteq \mathbb{N} \times \mathbb{N}$ be the pairs of depths and search stages the agent can afford under its entitlements. Under the sufficiency preorder $B \preceq B'$ assume $E(B) \subseteq E(B')$, with reusable capabilities held fixed or strengthened. An acquisition can change B in an incomparable way. Paying for a slot may make earlier purchases unaffordable even though the slot increases a specific attainable depth. We therefore avoid identifying growth with an order on every coordinate of holdings. A theorem about eventually attaining all stages uses a suitable ordered service or a net surplus that can be devoted to the relevant dimension.

Write $V(B)$ for the supremum of a stated objective over feasible plans given B, including the plan to do nothing. When $B \preceq B'$ and every plan feasible under B remains feasible with the same payoff under B', monotonicity gives $V(B) \le V(B')$. This is a comparison of entitlements prior to paying for an acquisition. It gives no inequality between the net utility of buying and the utility of declining.

Proposition 17.1 (Value of a priced observation)**.** *Suppose one agent has a finite action set, uncertain state X, and payoff $u(a, X)$ with finite expectations. An optional observation Y permits choosing an action after seeing Y. If the conditional expectations are defined and an optimal action can be selected measurably,*

its gross value is

$$\mathbb{E}\Big[\max_a \mathbb{E}[u(a,X) \mid Y]\Big] - \max_a \mathbb{E}[u(a,X)] \;\geq\; 0.$$

It is rational for an agent maximizing this expected payoff to buy the observation at price p exactly when its gross value is at least p, when a tie is resolved in favor of buying and the price enters additively.

Proof. A constant action is one of the policies available after observation. Taking the conditional expected payoff of an optimal constant action and then averaging yields the value without observation. The conditional maximum dominates that choice at every Y where the expectations are defined. A measurable maximizing selector realizes the upper expression. Subtract the additive price and compare with the value without purchase. □

If the observation is a deterministic exact finite prefix of a non-computable object, the proposition can be used after specifying an uncertainty model on such prefixes. The computability result does not itself assign a probability law. For example, if an agent's feasible action and payoff are independent of a particular higher bit, an observation revealing that bit can have zero economic value at every positive price. Conversely one bit can determine which of two commitments is realizable and have a substantial value. Informativeness depends on the downstream decision problem.

17.2 Growing search service from negotiated surplus

A strong but explicit way to obtain cofinal search is to require repeatable trades with a positive net contribution to a designated reusable compute service. Suppose that after the k-th completed trade a nonnegative integer q_k records the available service quota. Let c_k be the amount charged to obtain the trade and g_k the amount credited on settlement, measured in this service. The recurrence is

$$q_{k+1} = q_k - c_k + g_k$$

when the guard $c_k \leq q_k$ is met. Other resource coordinates may also be used, with separate guards. Let $C(n,s)$ be a finite sufficient quota for a requested observation, including the agreed verification rights. An agent may use an earmark for observations so that spending on a request does not consume the entire service reserved for the next trade.

Theorem 17.2 (Cofinal acquisition under persistent surplus)**.** *Suppose a controller can maintain a sequence of enabled trades with the following properties.*

Every trade completes in finite time under the assumed scheduler. The resulting quota after all charges at that step satisfies $q_{k+1} \geq q_k + 1$. For each fixed (n, s), its observation can be completed whenever the quota reaches $C(n, s)$ and the policy allocates a slot to it. Suppose the policy serves requested stages at depth n in increasing order, after the necessary quota is reached, with charges accounted for in the quota guarantee or covered by a separate earmark. Then the completed observations at depth n are cofinal.

Proof. Inductively $q_k \geq q_0 + k$. For any fixed stage s, the finite threshold $C(n, s)$ is reached after finitely many completed trades. By induction on the order in which observation requests are served, all preceding requests have already completed after a finite prefix, possibly later than their own thresholds. The quota guarantee and the allocation rule then allow the request at stage s to complete. This holds at every finite stage and stages are increasing. Thus the completed observations are cofinal. The conditional progress theorem of Section 3 is one sufficient way to establish finite completion of each trade. □

The theorem does not assert that a bounded closed ledger can supply unbounded physical units. If the service quota measures a consumable resource with a fixed total in M, the positive-surplus hypothesis cannot hold indefinitely. An environment that replenishes compute time, an expanding network with explicit external issues, or a reusable service with increasing available capacity can satisfy an appropriate model. The ledger identity accounts for this source. A transfer changes who controls a resource; it does not generate units without an issue or a conversion rule.

Corollary 17.3 (Limit-correct observation after repeated trades)**.** *Under Theorem 17.2, a limit-computable bit object has eventually exact completed observations at each served depth. This conclusion holds for the extension-class family using its finite prefix generator and test at the required depths.*

Proof. The cofinality theorem supplies stage indices tending to infinity. The finite observations stabilize at each fixed depth by Theorems 12.2 and 12.3. Every completed stage after that depth's stabilization threshold has the exact description. □

Each conclusion has a different failure mode. The quota threshold can remain beyond the holdings, the controller can choose not to spend on observation, the scheduler can fail to complete a trade, or the target bits can lie above the limit-computable class. An assertion that an autonomous agent is sufficiently advanced has mathematical content here only when it specifies access to the service and enough completed stages at the desired depth.

17.3 No effective network extracts a stronger oracle from its own computation

The following observation makes the boundary between computation and external information explicit.

Proposition 17.4 (Computable histories and definite answers)**.** *Fix finitely described effective controllers, an effective initial state, and a computable environment and scheduler. Suppose that on each query e, a designated output eventually returns a definite bit and the bit is guaranteed to equal $b(e)$. Then b is computable. If the query instead receives a computable stream of provisional bits guaranteed to stabilize to $b(e)$, then b is limit computable.*

Proof. Simulate the network on input e until the definite output appears. The hypothesis guarantees termination and correctness of this algorithm. For the second assertion simulate the first s steps, or the first s completed output events if the stream is guaranteed infinite, and record the latest provisional bit with a fixed default when none has appeared. This is a total computable approximation and stabilizes to the promised answer. □

The scheduler assumption covers computable ways of allocating a growing number of workers, provided their descriptions and state updates are uniform. A run with arbitrary nondeterministic choices can encode an arbitrary sequence in those choices. That is an input model with more information. A correctness guarantee over every allowed run would in particular have to hold on an allowed computable run if one exists. Physical access to additional information is a different resource from finite speedup.

17.4 Converting time and depth into one budget

A concrete evaluator receives a finite balance $B \in \mathbb{N}$ for its local computation. Let $C(n, s)$ be a computable scalar upper bound for the cost of producing its hypothesis descriptor. It need not bound the cost of all queries on the descriptor. If the agent wants a fixed class of tests, include their cost separately in C. For example, a bit test on the split descriptor can inspect the recorded bits, while a representation-independent test may enumerate Kan simplices. A test for the extension-class family can enumerate maps from Q_m, which has finite but potentially large cost.

Suppose $C(n, s)$ is monotone in s and grows without bound in s for each n, with at least one affordable stage when B is sufficiently large. The maximum affordable stage

$$S(n, B) = \max\{s : C(n, s) \leq B\}$$

is computable on the set of such budgets, by searching until the next stage exceeds B. For fixed n, $S(n, B)$ tends to infinity as B tends to infinity. At a limit-computable target this yields eventual exactness in the budget parameter. If a computable budget function $B(n)$ always yielded an exact halting prefix using this evaluator, $S(n, B(n))$ would supply a forbidden computable choice of exact stage. Thus replacing step count by a computable monetary or quota translation does not remove the lower bound.

For an arbitrary oracle degree, a purchase contract can promise an exact stage supplied by a source of that information. A one-time contract may deliver a code and a proof verified under trusted input assumptions. A uniform service for all depths implements the oracle requirement of Theorem 10.4 or Theorem 11.4. An ordinary provider can still perform the computation of applying the finite descriptor once it has been supplied. The ideal network can therefore incorporate objects beyond effective provision without treating every executable action as an oracle call.

17.5 Coalitions and the order of decisions

Several participants can combine resources to obtain an observation that none can buy alone. The minimum feasible split of a scalar cost c among participants with spendable holdings h_i exists if and only if $c \leq \sum_i h_i$ in the discrete fungible model. For necessity sum the debits. For sufficiency allocate debits sequentially by charging each agent the minimum of its balance and the remaining cost, which reaches zero by the inequality. With separate kinds apply the construction coordinatewise. Transfer restrictions or non-transferable capabilities can invalidate this reduction; they must be included in the admissible split relation.

A feasible split does not force agreement to share its cost. Let v_i be each participant's value for a successful observation and suppose positive values and additive costs are common knowledge, with the option of refusing. If each i is to pay p_i, the inequalities $p_i \leq v_i$ express individual rationality under a zero outside option. With unbounded divisible monetary transfers, a nonnegative split with $\sum_i p_i = c$ exists exactly when $c \leq \sum_i v_i$. The proof is the same allocation argument over nonnegative reals. With balances also imposed, replace v_i by $\min(v_i, h_i)$. Strategic free riding can still prevent formation of such a coalition if agents benefit after refusing. The negotiated rights and visibility of the result determine that deviation. This is another instance of the separation between feasibility, incentive compatibility, and eventual availability.

18 Local contracts and higher descent

18.1 A site for contextual information

An agent often sees a contract only within a context: one region of a network, a particular time window, or a subsystem over which it has reading rights. A category $\mathcal{C}$ of contexts can record restriction arrows. A Grothendieck topology specifies which families of smaller contexts jointly cover a larger one. The semantic model is the ∞-topos $\mathcal{X} = \mathrm{Shv}(\mathcal{C}; \mathcal{S})$ of sheaves of spaces. This is an optional refinement of the global model. It does not infer a topology from the resource order alone. In particular two entitlements may jointly cover a service specification even when neither has more resource in every coordinate.

A contract in this model is an object B of $\mathcal{X}$ and a witness family can be given by a map $W \to B$. A point over a context U is a section over U. A family of compatible local sections includes comparison paths on overlaps, higher comparisons on multiple overlaps, and the further coherences dictated by descent. In the special case of sheaves of sets the first equality condition determines all higher comparisons. For a general sheaf of spaces the higher data are part of the input to a gluing problem.

Let $U_\bullet \to U$ denote the Čech nerve of a cover, represented in the topos by the iterated fiber products of the covering object over U. For a sheaf E write $E(U)$ for its section space. Descent is the equivalence

$$E(U) \simeq \lim_{[k]\in\Delta} E(U_k).$$

The right side retains the entire cosimplicial compatibility diagram. The Čech nerve is used for a cover. A statement about descent for arbitrary hypercovers requires the appropriate hyperdescent hypothesis [23, 24]. In particular this paper does not exchange every form of descent with Postnikov convergence in an arbitrary ∞-topos.

18.2 Descent of open negotiation

Let P, Q, B be sheaves of spaces with legs $P \to B \leftarrow Q$. The homotopy pullback in the topos is again a sheaf. It is evaluated as a homotopy pullback at each context because the inclusion of sheaves into presheaves preserves limits, and presheaf limits are computed objectwise.

Theorem 18.1 (Descent commutes with a seam)**.** *For any cover $U_\bullet \to U$, the type of joint witnesses on U is naturally equivalent to the limit of the types of*

local joint witnesses:

$$P(U) \times^{h}_{B(U)} Q(U) \simeq \lim_{[k]\in\Delta} \left(P(U_k) \times^{h}_{B(U_k)} Q(U_k) \right).$$

The comparison preserves the projections and the designated seam path. It also commutes with sequential limits of such witness diagrams.

Proof. Apply descent to each sheaf P, B, Q. Form their homotopy pullback and interchange the two limits. The result is the stated diagram, because the homotopy pullback is itself a limit. Naturality of the projections follows from this universal property, and the seam path is part of the cone. For a sequential tower, the two limits are both right adjoints to constant diagram functors in their respective indices. The iterated limits are equivalent to the limit over the product index, which again commutes with the pullback. The comparison may also be proved using the explicit sequential path formula of Section 7 after taking sections. □

This theorem concerns compatible local witnesses. A list of local inhabitants without the comparison data does not satisfy its hypothesis. Even if every pair of local choices can be related, different choices of pairwise relations can fail the triple condition. A contracting choice on each chart can also be incompatible with a specified contract on the overlaps. The following examples separate the first obstruction from a genuine higher one.

18.3 A discrete torsor with no global choice

Cover a circle by two connected open arcs U, V whose intersection has two connected components. Take a locally constant two-element family, trivialized on each arc. On one overlap component the transition is the identity. On the other it adds 1 in $\mathbf{C}_2$. The gluing defines a two-sheeted cover. A section on each connected arc is a constant element in its chosen trivialization. To glue two values a, b they must satisfy both $a = b$ and $a = b + 1$. There is no solution. Local witnesses exist everywhere. A coherent global agreement does not exist.

The obstruction can also be expressed by monodromy. The path around the circle acts on the two-element fiber by the nontrivial permutation. A section of the associated locally constant family would give a fixed point. The witness type is the homotopy fixed point type of the action of the loop group on that fiber, and here it is empty. This gives a resource interpretation in which two jurisdictions issue local names for one capacity reservation. The protocol may permit either name in each jurisdiction, while a single globally reusable name would need to be invariant under both changes of trivialization.

18.4 A non-split extension as a coherence obstruction

Let $q : \mathbf{C}_4 \to \mathbf{C}_2$ be reduction modulo two. Consider $Bq : B\mathbf{C}_4 \to B\mathbf{C}_2$. A section up to homotopy would induce a section of q on fundamental groups. A generator of $\mathbf{C}_2$ must be sent to an element of order at most two and odd parity. No such element of $\mathbf{C}_4$ exists. Since both groups are abelian, changing the chosen basepoint paths has no effect on this calculation.

There is a concrete description of the failed coherence. Take the representatives $s(0) = 0, s(1) = 1$ in $\mathbf{C}_4$. Every element of $\mathbf{C}_2$ has a chosen lift. The discrepancy

$$c(a, b) = \frac{s(a) + s(b) - s(a + b)}{2} \pmod 2$$

is 1 at $(a, b) = (1, 1)$ and 0 otherwise. Addition in the numerator uses integer representatives, with $a + b$ in $\mathbf{C}_2$. The associativity of addition makes c a normalized 2-cocycle with trivial action. To see this, express both parenthesizations of $s(a) + s(b) + s(d)$ using the lifts of partial sums and reduce the two carry terms modulo two. The resulting identity is

$$c(a, b) + c(a + b, d) = c(b, d) + c(a, b + d).$$

Changing the section by $2\lambda(a)$ changes c by a coboundary. For a normalized λ the coboundary at $(1, 1)$ equals $\lambda(1) + \lambda(1) - \lambda(0) = 0$ in $\mathbf{C}_2$. Thus the class of c is nonzero. Individual comparisons can be lifted, but the lifts cannot satisfy the triangle of composition. This obstruction is the extension class in $H^2(B\mathbf{C}_2; \mathbf{C}_2)$.

In a negotiation the base labels can encode equivalences of proofs issued under a two-element namespace, and the total labels can encode the permitted authorization changes. A scheduler can use any one lift of a comparison at one occurrence. A globally reusable procedure that must compose the same way across all comparisons is a section in the homotopical sense. The example gives a rigorous difference between existence of separate lifts and a compositional authorization plan. Its calculation takes place in classifying spaces of finite groups and is fully effective as a presentation.

18.5 A general obstruction and the space of resolutions

Let A be an abelian group, $d \geq 1$, B a space of contexts, and

$$k : B \to K(A, d + 1)$$

a map. Define $E = \mathrm{fib}(k)$ at the basepoint, with projection $E \to B$. The type of sections over a map $b : U \to B$ is

$$\prod_{u : U} (k(b(u)) = *) \simeq (k \circ b = \mathrm{const}_*) \quad \text{in } \mathrm{Map}(U, K(A, d + 1)). \tag{14}$$

The first type describes sections by Proposition 11.1, and the displayed equivalence is function extensionality. Thus the obstruction to a section is the class $b^*[k] \in H^{d+1}(U; A)$, using representability of ordinary cohomology. If this class vanishes, choosing one section identifies the space of sections with $\Omega \operatorname{Map}(U, K(A, d+1)) \simeq \operatorname{Map}(U, K(A, d))$, with loop taken at the constant map. The identification composes a null path with the inverse of the chosen null path to form a loop, and the inverse action concatenates that loop with the chosen path. The choice of a section is part of this equivalence. The set of components of the section space is then a torsor for $H^d(U; A)$, and higher comparisons persist in lower cohomological degrees where the mapping space has nontrivial higher homotopy.

Equation (14) makes the relation with the earlier truncation theorem transparent. The path to the basepoint is data in degree one higher than the choice of a component of its loop space. More generally the section space over a probe of dimension d' is subject to the depth calculation in Section 9. The two uses of dimension are distinct: $d+1$ is the coefficient degree of the obstruction, while d' is the shape of the context on which a lift is required.

18.6 Restriction and realized observation

There are two different operations that can be called restriction. For a map $V \to U$, the section map $E(U) \to E(V)$ is induced by precomposition. Taking sections over a fixed U is representable and preserves all limits. For a geometric morphism of ∞-topoi, the inverse-image functor preserves finite limits and colimits, but it need not preserve arbitrary countable limits. Hence a change of topos carries the finite seam construction to the corresponding seam in the target. A claim that it carries realization of an infinite observation tower to the limit of the transported tower requires a separate countable-limit comparison.

The relevant instance of Postnikov convergence is also local to the model. In spaces every X^b and Y^b is recovered by its tower. In a general topos an object can have the same truncations as another while comparison with the Postnikov limit is not known to be an equivalence. One can restrict attention to objects for which the comparison is an equivalence, or explicitly work with the tower as the semantics. This choice affects ideal reconstruction and leaves the finite-limit gluing theorems available. An agent's ability to navigate the context site is an additional access datum; it does not follow from a general categorical descent theorem.

19 Resource grades and contextual refinement

19.1 Charging a composite

Let $M = \mathbb{N}^R$ and regard M as a discrete space. A resource-graded span $U \leftarrow P \to V$ includes a cost map $c_P : P \to M$. The cost is invariant under witness paths. This assumption is appropriate when a path changes a proof of the same charged execution. If a faster execution of a protocol has different charges, it can belong to a different component of the graded witness space or be compared by an explicitly cost-changing refinement. A semantic identification of the ungraded witnesses alone does not fix their charges.

Define the composite cost on $P \times^h_B Q$ to be $c_P(p) + c_Q(q)$. The identity span has cost zero. The associator in Proposition 6.1 preserves the total cost because its forward and inverse maps retain all local witnesses, and addition is associative. The proof of the unit law uses the same zero contribution. If disjoint resources are used in parallel, the product span can be graded by the sum of the two cost maps. The interchange equivalence of Section 6 preserves that grade by commutativity and associativity. When a protocol shares a fixed verification charge across several recipients, the charge must be assigned to one shared witness rather than copied into each local cost map. A naive sum then models a different transaction.

For $a \in M$, let $P_a = \mathrm{fib}_{c_P}(a)$ and $P_{\preceq B} = \sum_{a \preceq B} P_a$. Since M is discrete, the first definition selects the subspace of exact cost a. The second is the subspace affordable under a cost bound B, holding the other entitlements fixed. This cost bound does not assert that a physical reservation is available, which is a condition of the operational ledger.

Theorem 19.1 (Graded gluing)**.** *For graded spans P, Q composable over V, there is a natural equivalence over the outside endpoints*

$$(P \times^h_V Q)_{\preceq B} \simeq \coprod_{a+b \preceq B} (P_a \times^h_V Q_b).$$

When R is finite, the indexing set of splits on the right is finite. Given finite-stage codes for the apices, boundary maps, and cost maps, the affordable composite has a uniform finite-stage code.

Proof. A point on the left consists of p, q, a seam path, and the proposition $c_P(p) + c_Q(q) \preceq B$. Assign $a = c_P(p), b = c_Q(q)$. This gives the indexed pair of exact-cost fibers and the same seam path. Conversely project each fiber witness to its local point and substitute the cost equalities into the bound. These maps are inverse by path induction on the cost equalities. They preserve the boundary maps because they do not change p, q. Every coordinate of a, b is bounded by the

corresponding coordinate of B, so the indexing set is finite. For the effectivity assertion obtain each exact fiber by matching against the singleton a or b in the discrete cost space. Equivalently filter the simplices by the computed discrete grade. These are finite-stage presentations: their homotopy types are unions of components of the apices. Apply Lemma 4.2 to each seam and take the finite disjoint union. All operations are uniform in the code and the supplied bound. □

There is a presentation issue when M is infinite. The target cost map is a map into a discrete computably encoded set. One need only be able to compute its value on a simplex, rather than list the entire target at each degree. Filtering by that value preserves the simplicial operators, and a union of components of a Kan complex is Kan. Thus the theorem uses finite-stage codes for the witness spaces and decidable comparison of individual grades.

Example 19.2 (A local bound is not a split)**.** Suppose two computations each use four time units and an agent has a total of six. Both local apices survive restricting each to cost at most six. The composed witness has cost eight and is absent from the right side of Theorem 19.1. The correct local tests quantify over a split of the common bound. This distinction also appears when a provider plans two sequential verifications in one expiring time interval. The sum rule applies when the model charges a single sequence of time quanta, while a parallel model with two independent processors can use a different resource grading.

19.2 Refinement through public interfaces

A cost-nonincreasing refinement between spans with the same endpoints is a map $F : P \to Q$ over those endpoints, together with $c_Q(Fp) \preceq c_P(p)$ at every point. Boundary comparison paths are part of the data of a map over the endpoints. Such a refinement induces a map $P_{\preceq B} \to Q_{\preceq B}$ for every bound, by transitivity. An effective refinement additionally supplies a computable translation of witness codes when the source and target are presented. To obtain an operational guarantee the cost of running this translation must be included or separately budgeted.

Proposition 19.3 (Contextual simulation)**.** *Cost-nonincreasing refinements are closed under sequential gluing and independent parallel composition. In every finite context formed by those operations, replacing a component by its refinement carries each affordable composite witness to an affordable witness with the same exposed endpoints up to the specified paths.*

Proof. For sequential gluing, a pair of refinements sends the local witnesses p, q to their images. Translate the seam path by the two given boundary comparison

paths, as in Section 5. The resulting point has total charge bounded by the old total because addition is monotone in each coordinate. It is over the outside endpoints by the comparisons on the remaining legs. For parallel composition, use the product of the maps and their endpoint comparisons, with the same monotonicity argument. Identity refinements have equality of cost and trivial endpoint comparisons. Induct on the syntax of the finite context to obtain the last assertion. The associativity and substitution comparisons ensure the construction agrees with equivalent rebracketings of the context. □

The refinement is directed. For example, replacing a verifier with a cheaper implementation may preserve every old feasible witness while admitting new proofs and additional actions. This does not yield a reversible equivalence of strategies, since new actions can change regrets. A directed refinement theorem for feasibility and a bidirectional strategic equivalence theorem answer different questions. The latter requires the deviation coverage of Section 16 or the exact correspondence there.

19.3 What tests can characterize a span?

Fix outside endpoints U, V and write $J = U \times V$. Regard a witness span as an object of the slice $\mathcal{S}_{/J}$. A test consists of a space T with a map $T \to J$. Its response type is

$$\mathrm{Map}_{\mathcal{S}_{/J}}(T, P).$$

It records all coherent lifts of the test along the witness map. For example, a one-point test asks for a single agreement with specified endpoints, a sphere tests a family with a fixed boundary condition, and a finite higher probe can distinguish the two Postnikov blocks of Section 11.

Theorem 19.4 (Contextual determination)**.** *Suppose two spans over J have naturally equivalent response types for every test in the slice. Then the spans are equivalent over J. Conversely an equivalence over J induces such a natural equivalence. The conclusion can be applied to the subspaces of each exact cost if the tests carry that grade as another boundary coordinate.*

Proof. The response functor is the presheaf represented by the apex in the slice. The Yoneda lemma identifies maps between the representable functors with maps between the apices. A natural equivalence and its inverse yield maps whose composites are homotopic to the identities, so the apices are equivalent in the slice. Conversely use composition with an equivalence and its inverse. For a graded span take $J \times M$ as the boundary, or take the fibers at each grade before applying the statement. □

A test by a single point does not determine the required naturality. Separate identifications of fibers at each context can fail to commute with transport. The torsor of Section 18 and the trivial two-element family have equivalent fibers, yet their global section spaces differ. In a network the meaning of an interface depends on how agreement proofs change along contextual morphisms, not solely on what the agent sees in a single snapshot. If only finite probes are allowed, the Yoneda argument as stated does not apply to all apices, because the apex itself may not be an allowed test. The pro mapping formula of Section 14 and the explicit detecting probes supply results for specific restricted test classes.

19.4 A budgeted test is a joint problem

Suppose a tester supplies an outside request $T \to J$ and wants an affordable response. It can first restrict the apex to the allowed cost and then take its slice mapping space. This requires every point of the test to have a witness with the allowed cost, consistently over the test. It is a different question from summing costs over all points of T. In a verification contract, one requirement may charge for maintaining a reusable routine parametrized by T, while another charges for executing it at each requested input. The grading must say which resource is consumed. If a family has cost measured by its implementation as one program, that program and its grade should be part of a new witness type.

One can price the comparison itself. A controller with a reserve B can transport a witness of cost a along an effective refinement whose compilation or translation cost is at most c, provided $a + c \preceq B$ and the other guards hold. If the new witness has smaller execution cost, it may be cheaper after translation, but the lemma on contextual simulation only compares the witness grades. Proving that the translator is operationally affordable adds a graded transition for the translation. This illustrates how the semantic laws can be composed with the ledger without treating semantic equivalence as free computation.

19.5 Limits of graded spans

If a tower of spans has cost maps into a constant discrete M, commuting with transition maps, the limiting span has a well-defined grade. A compatible family of witnesses has the same grade at every stage, because applying the cost maps to its coherence paths gives identities in M. Consequently the exact grade fiber of the limit is the limit of the grade fibers, by the same finite-limit interchange as in Theorem 7.2. For a bound B take the fiberwise pullback along the discrete inclusion $\{a : a \preceq B\} \hookrightarrow M$. The operation commutes with realization because that inclusion is used as a constant diagram and finite limits commute with the inverse limit.

If instead stage costs can increase with depth, the cost maps do not define a map into a constant discrete M. The operational question is then whether a sequence of charges can be sustained, or whether it has a common bound. A monotone cost sequence in $\mathbb{N}^R$ has a finite common bound exactly when every coordinate is bounded, and it is then eventually constant when R is finite. This elementary conclusion does not construct a computable upper bound from a presentation of the sequence. For a model of increasing service commitments it is appropriate to keep cost trajectories. The fixed-grade interchange theorem and the cofinal service theorem address these two distinct regimes.

20 Negotiation patterns

20.1 Buying compute with a verification option

A buyer A needs q units of computation during an interval J. A provider P will supply the units at price p if an independent verifier V supplies an admissible completion receipt. The verifier requires payment f and time to check the specified output. A direct implementation has three stages. First the parties record a bundle describing q, J, p, f and the accepted receipt type. Then they reserve the required monetary and service balances. Finally the settlement rule checks the receipt and closes the agreed bundle of reservations. If the deadline passes under the release rule, the spendable amounts return to their owners. Consumption of the time slot can be recorded in the retirement balance, or the slot can be modeled as an expiring right, with its expiry accounted for separately.

The buyer–provider component has apex P_1 consisting of a compatible offer, a buyer acceptance, and a feasible reserved computation plan. It maps to a receipt specification B by sending a plan to the verifier's required receipt type and interval. The verifier component has apex P_2 of executable verification plans, with a map to B exposing its promised specification. The joint witness is a point of $P_1 \times^h_B P_2$. The seam path lets the settlement rule transport the verifier's proof to the buyer–provider contract. If the rule accepts all witnesses produced by that transport, delivery of the proof while the reservations remain open gives the enabled settlement transition. This last sentence adds the operational guard and scheduler hypotheses to the higher existence claim.

Suppose receipt specifications can be relabeled by an abelian group G of proof-interface changes, and $B = BG$ over fixed price and intervals. With constant pointed legs the matches of the two chosen plans form G. If each relabeling is valid and the spend is keyed by the same settlement name, those matches give multiple valid proofs for one authorized distribution. Ledger safety still permits one successful closing effect. If the verifier's interface is instead $K(G, 2)$, the

matches form $K(G,1)$. They now lie in one component while retaining a group of proof self-comparisons. Both cases show why a change in witness multiplicity does not change the amount of resource allocated.

The strategic condition can be made explicit. Let v_A be the buyer's value for the completed computation, c_P the provider's cost, and c_V the verifier's cost. If the provider pays f to the verifier and receives p from the buyer, then with zero outside options individual rationality is

$$p \le v_A, \qquad p - f \ge c_P, \qquad f \ge c_V.$$

With divisible nonnegative transfers, feasible prices exist exactly when $v_A \ge c_P + c_V$, assuming the quoted costs are nonnegative and ignoring balance constraints. The implication from prices follows by addition. Conversely take $f = c_V$ and $p = c_P + c_V$. A balance constraint adds the guards of the reservation model. A nonzero outside option adds the corresponding required surplus to each inequality. None of these price inequalities implies that the verifier's promised interval meets the buyer's seam. The seam is an independent feasibility test.

20.2 Negotiated visibility and a higher verification obstruction

Suppose a verifier posts a finite public description of how each local proof transformation works. A buyer needs a single plan valid across a finite family of configurations Q_m as in Section 11. The contract requires a map $Q_m \to P_m$ inducing the nontrivial element on the designated m-sphere. A seller with the split block can give the plan by restriction of its section. A seller with the twisted block has the same local homotopy groups and cannot give such a plan. Merely comparing the published groups leaves the two offers indistinguishable.

If the offer is subject to an exact finite diagnostic, the buyer can test it by the algorithm of Proposition 11.3. An agent who owns only depth m of the Postnikov data sees the same lower stage of either block. The full block is exposed by depth $2m - 1$. A budget that buys access to the full block and the finite diagnostic makes the previously ambiguous choice decidable for that offer. For the family of infinitely many blocks, uniformly supplying that exact service has the oracle degree of the hidden bit sequence. There is no contradiction between the finite decision and the uniform lower bound. The finite decision starts from a valid code for a particular block.

A provider can accept a binding contract that pays only when the plan is produced. If no plan exists, the resource reservation must be released under the agreed deadline. Under the failure and fairness assumptions in Theorem 3.4 this protects the buyer from paying for a non-lift. It does not reimburse the time already spent searching unless the settlement rule and the available ledger

explicitly account for such a cost. A higher witness theorem establishes which proof can exist. It does not assign liability for discovering the answer.

20.3 A coalition building a finite observation

Consider three agents A, B, V. Agents A and B each need a finite observation at depth n for a separate downstream contract. Agent V has a reusable evaluator but needs a joint deposit covering one execution at stage s. The result is nonrival evidence. The contract distinguishes transferable funds, the evaluator's fuel, and the right of each client to see the resulting code. Once the code is delivered, its duplication costs no additional execution of the evaluator. Distribution still depends on the agreed access rights. The two clients can share a one-time charge without duplicating the evaluator's capacity.

Let the full charge be c and the clients' spendable holdings be h_A, h_B. A split exists when $c \leq h_A + h_B$. For example, with integer amounts $c = 7$, $h_A = 4$, $h_B = 5$, any split with $2 \leq p_A \leq 4$ and $p_B = 7 - p_A$ meets those balance guards. The allocation is not unique. If the values are $v_A = 3, v_B = 5$ with zero outside options, individual rationality further imposes $p_A \leq 3, p_B \leq 5$. The admissible integer prices are then $p_A = 2$ or 3. The ability to inspect a shared code can still create a deviation if one client receives the information after declining to pay. Modeling the result as a restricted reading right changes that deviation and should be reflected in the strategy sets.

The composite witness has seams for the funded deposit and the returned code. If the target is X_n^H, the finite computation of $h_s|_n$ always returns a valid hypothesis code. A contract promising a hypothesis can settle based on the bounded computation trace. A contract promising the exact halting prefix cannot use the same trace as a complete exactness certificate. A positive halt may be witnessed, but a zero in the word records only the simulation outcome. A provider could supply a trusted exact word under a different information model. The distinct obligations therefore require distinct contract types.

Suppose the joint evaluator is repeated at cofinal stages. After the stabilization time the code returned by every sufficiently advanced evaluation is the exact code at that depth. If the clients use each report provisionally and reserve irreversible decisions for a verifiable positive witness, they preserve the advertised guarantee of the provisional contract. If they interpret every zero as an exact negative and settle an irreversible contract immediately, an eventual later positive report cannot repair the old effect. A design that needs exact negatives must expose either a restricted decidable input class or a source of definitive answers. This is the operational manifestation of Theorem 12.1.

20.4 An alternating-offer benchmark

A simple strategic benchmark isolates a price negotiation without hidden verification information. Two participants A, B divide one unit of known transferable surplus. They alternate proposals, beginning with A. A respondent can accept or reject. After rejection the proposer changes and both parties discount one round, with factors $0 < \delta_A, \delta_B < 1$. Infinite disagreement has payoff zero. Offers and responses are observable. The model uses divisible transfers, no additional resource acquisition between rounds, and the tie convention that a respondent accepts an offer equal to its continuation value. This is a form of the alternating-offer model [30].

Let a denote A's share in its proposer state, and b denote B's share in its proposer state. Consider stationary strategies in which each proposer offers the other participant its discounted value of being next proposer, and each responder accepts an offer at least that value. The equations are

$$a = 1 - \delta_B b, \qquad b = 1 - \delta_A a.$$

Solving gives

$$a = \frac{1 - \delta_B}{1 - \delta_A \delta_B}, \qquad b = \frac{1 - \delta_A}{1 - \delta_A \delta_B}.$$

The denominator is positive. Each share lies strictly between zero and one. The given proposal is the best accepted offer for its proposer, since increasing the respondent's share decreases the proposer's share. When A's proposal is rejected, its continuation payoff under the stated strategies is $\delta_A(1-b) = \delta_A^2 a < a$. Similarly B cannot gain by inducing rejection. The responder's threshold is optimal against the stated continuation. These inequalities cover the possible one-step deviations at each continuation state.

For completeness, the local inequalities imply no profitable multi-step deviation. Given the other participant's strategy, the stationary profile gives a value at each proposer state and each responder state. For any immediate unilateral choice the value after applying the discount and the next-state value is at most the prescribed value, by the preceding threshold and best-accepted-offer calculations. Iterating this inequality along a deviating plan bounds its first T rounds plus the discounted tail by the prescribed value. The tail vanishes as $T \to \infty$ because payoffs are bounded. Thus the prescribed response is optimal in every continuation state and the profile is subgame perfect. A claimed uniqueness theorem would require controlling other strategies and is unnecessary for this benchmark.

The higher model changes the benchmark when feasibility depends on proof witnesses. If all feasible divisions in the benchmark have a matching witness

and all deviations in the strategy proof transport to legal deviations, the same payoff calculation applies by Proposition 16.1. If a low-price offer requires a witness that exists only after acquiring a verification slot, the set of accepted and executable divisions can shrink or change. The algebraic shares then describe a game with different actions. Incorporating that slot into a staged game requires the additional state coordinate and the admissible transitions modeled in Section 17.

20.5 Observations that arrive after commitment

Here is a failure caused by timing rather than depth. Two sellers offer the same service. The first quote expires at time t_0. The second has a longer deadline but needs a proof whose matching path becomes visible only after time t_0. An ideal higher observation can compare both complete witnesses. The effective buyer at time t_0 has a choice between accepting the first quote and continuing to search. It does not have a choice between two live, verified offers. To use the ideal comparison one must either model an option extending the first quote, attach a cost to delay, or restrict the comparison to realizable strategies at each time.

The interplay can be stated as an instance of Theorem 16.5. If the current approximate strategy set contains the first choice but not the second, an ideal deviation using the second has no guaranteed representative among approximate deviations. A payoff bound on the first choice cannot replace that coverage hypothesis. Conversely if the ideal comparison ignores expiry, the realization of the proposed deviation fails at the operational guard. The correct simulation relates strategy histories at their actual observation times and preserves the resource and authority checks of the ledger.

20.6 A conditional service guarantee

Put the examples together. Let a chain of agents expose a fixed graph of spans for a finite-depth service. Suppose each participant has a bounded evaluator and a ledger-backed reservation. Suppose their finite diagrams stabilize up to equivalence at depth $n + 1$, and the contract asks for an invariant property of the resulting agreement through depth n. Then the stagewise computation of the composed graph eventually produces the correct n-type, by Corollary 8.3 and stability of homotopy pullbacks under equivalence of diagrams. If each local hypothesis is given by a uniform finite-stage code, Lemma 4.2 supplies a uniform code for the stagewise graph match. Further effective evaluation of the requested property requires a code for that property, as in the finite-probe tests.

If the contract also promises an eventual successful spend, additional hypotheses are required. There must be an available admissible witness, the agents must

choose and deliver it, and the settled bundle must remain enabled under the specified scheduling rule. Theorem 3.4 then proves closure with that outcome when the settle rule is the only admissible closing rule on the relevant tail. If strategic stability is also desired, the feasible profile and its deviations must be covered by the strategic data. This service guarantee therefore states its semantic, computational, operational, and strategic components separately, with an explicit implication from one layer to the next where it is justified.

Part V
Formal development

21 Kernel-checked proofs in Lean HoTT

21.1 The kernel and the library

The companion uses the homotopy type theory mode of Lean 2, with the upstream HoTT library at the pinned source commit listed in the bibliography [19, 7]. That mode has proof-relevant identity types. This matters here because an equality of two interface records is a space of matching witnesses. A foundation in which all equality proofs are definitionally interchangeable would require an explicit external model of paths to express the same theorems. The supplied `.hlean` modules use the proof-relevant kernel directly.

The library supplies identity elimination, pathovers, cubical squares, dependent sum and product equivalences, function extensionality, and higher inductive truncation. Its Eilenberg–Mac Lane development supplies `EM.EM`, the loop equivalence `EM.loop_EM`, and homotopy group computations [7, 5]. The companion proves the negotiation-specific laws by combining these constructions. For example, a `square` is a type with witnesses, and converting it to a commutative equation between composites preserves the higher information. The proofs use explicit equivalences with inverse homotopies, as in the paper.

21.2 The sequence of proof modules

The basic associator uses the library's pullback record, with fields for the two witnesses and a path between their images. The two functions simply reassemble the same tuple, so induction on the records proves the inverse laws. The unit uses based path induction on the path from a fixed $f(a)$ to the extra endpoint. The proof can eliminate that path even when its starting point is a composite expression. This is the synthetic form of the contractibility of a pointed singleton

Module	Checked declarations and their role
`Agreement`	Units, associativity, substitution, transport of certificates, and the reduction of truncation to the path type of the mediator.
`Tower`	Proof-relevant sequential limits, the dependent equivalence for paths between coherent families, and the rebase equivalence for comparison squares.
`Gluing`	The full equivalence between the limit of levelwise matches and the match of the limiting types.
`Observation`	The canonical cospan truncation equivalence, a pointed higher loop calculation and the Eilenberg–Mac Lane group calculation.
`Prefix`	Reconstruction of the product from its prefix tower, specialized to a family of Eilenberg–Mac Lane types.
`Sharpness`	Contractibility of the shallower composite and nonexistence of an equivalence with the exact observation for a nontrivial abelian group.
`Obstruction`	Equivalence of lift and nullhomotopy spaces, splitting over the base, and invariance of the section space of a truncation.
`Audit`	Dependency reports for the principal declarations after the proof modules have been loaded.

Table 1: Coverage of the formal companion. The code is organized by general theorems on types and towers.

used in Proposition 6.1.

The `Tower` module represents the compatibility family by a dependent product indexed by the natural numbers. A path of limit elements first decomposes into a path of their sequences and a pathover between the two coherence fields. Function extensionality changes the first path to a pointwise homotopy. Path induction reduces the pathover formula to the case of two coherent families over the same sequence. In that case function extensionality on the coherence field and the vertically degenerate square equivalence identify the remaining path with a family of squares. This is the precise synthetic realization of Lemma 7.1.

21.3 How the checked gluing proof is assembled

The program uses two presentations of a match. A library pullback is a record. The equivalent presentation used for the stagewise tower is a dependent pair whose base is $A \times C$ and whose fiber is $f(a) = g(c)$. A path in that dependent pair decomposes first into a base path and a pathover. Paths in $A \times C$ split into the two coordinate paths. The pathover over those coordinate paths is equivalent to a square in the mediator. The conversion is proved by induction on the base path and uses the same degenerate-square equivalence as the tower path formula.

The next step reorganizes a sequence of matches into its three rows, the two sequences of points and the sequence of seam paths. Each stage comparison produces one compatibility path for each outside sequence and one square for the seam. The comparison square as obtained from the stage match has the reoriented paths $\theta_f^{-1} \cdot \mathrm{ap}(r_B, p) \cdot \theta_g$. The `rebase_square` equivalence converts it to the square that characterizes a path between the images of the two limit points. In the proof, induction on the four boundary changes reduces the conversion to two equivalent forms of degenerate square. The middle seam paths are left as variables. As a result, the theorem holds in every homotopy degree.

The final sequence of equivalences is short once the reorganizations and path formulas have been proved. The excerpt shows that it is a composition of equivalences in the HoTT kernel. The complete source supplies the types of each reorganization.

```
tower_limit (match_tower f g)
    ~> separated rows and coherence
    ~> outside limit points and compatible seam paths
    ~> outside limit points and a single limit seam path
    ~> pullback (limit_map f) (limit_map g)
```

In the listing the arrows denote the indicated equivalences, and the intermediate labels describe the fields of their types. The actual declaration is `negotiation.gluing_limit`. Its proof starts with the rows equivalence, then applies the coherence equivalence to each fiber, packs the resulting outside limits, applies the inverse of the path characterization, and finally uses the library's characterization of a pullback by a dependent sum.

21.4 The checked truncation comparison

In `Agreement`, the mediator replacement in Theorem 8.1 uses the library equivalence between a truncated path type and the path type of a truncation one degree higher. To use this fiberwise, the proof rearranges the pullback as a dependent sum over two endpoint witnesses. The standard equivalence $\|\sum_z P(z)\|_n \simeq \|\sum_z \|P(z)\|_n\|_n$ is applied to that sum. Each truncated path fiber is then replaced by the path in the truncated mediator. This checks the part of the theorem in which the two outside spaces are unchanged.

The `Observation` module checks the comparison when both outside spaces are also truncated. It uses the following auxiliary equivalence for any type family P over $\|A\|_{n+1}$:

$$\left\| \sum_{a:A} P(|a|_{n+1}) \right\|_n \simeq \left\| \sum_{z:\|A\|_{n+1}} P(z) \right\|_n .$$

The forward direction applies the truncation unit in the first coordinate. For the inverse, at a point z construct the function $P(z) \to \left\| \sum_{a:A} P(|a|_{n+1}) \right\|_n$ by eliminating z. Its target family is $(n+1)$-truncated, since a function into an n-type is itself n-truncated. On a point of the form $|a|_{n+1}$ the function sends v to $|(a, v)|_n$. Eliminating the outer truncations and then the base truncation proves the inverse laws. Applying the equivalence once for each of the two endpoints proves the full comparison. The declaration `truncation_of_cospan_is_canonical` also identifies its forward function with truncation of the map on witnesses that applies the truncation unit to each endpoint and applies it to the seam path. The calculation on constructors extends to all inputs by truncation elimination.

The loop calculation uses an arbitrary abelian group and a natural number n. Taking two pointed singleton maps into $K(G, n+1)$, the code identifies their match with $K(G, n)$ using the library's loop equivalence. The higher group declaration records the identified π_n of $K(G, n)$.

The `Sharpness` module then proves the homotopical claim in Proposition 8.2. Truncating each input at n gives three contractible types. The proof of contractibility of their pullback uses the characterization by a dependent sum of path spaces and contracts each factor in turn. The actual pullback is equivalent to $K(G, n)$, which is already an n-type. An equivalence between the two observations would make $K(G, n)$ contractible. Its iterated loops and then its π_n would be contractible, contradicting the assumed noncontractibility of G. The formal result gives a function from a proposed equivalence to the empty type at every degree, including degree zero. The finite presentation of the $\mathbf{C}_2$ example is supplied by the cocycle construction in Section 10.

21.5 A split tower in the kernel

The `Prefix` module works with an arbitrary family X_n in one universe. Its stage zero is a lifted unit type. Its successor stage appends the next factor by a product, and the transition is the first projection. From a coherent choice, read the last coordinate of stage $n+1$ to obtain a point in X_n. Conversely build each prefix from a given full sequence. Reading a newly built prefix is definitionally the supplied coordinate. For the other composite, the proof constructs an equality of the original and rebuilt prefix by induction on depth. At the successor, the left component of the path combines the stored transition with the previous prefix comparison, and the right component is reflexivity. The square law for limit paths follows from the formula for the first projection of a path in a product. This proves equivalence of the compatible choice type and the product of all factors.

Specializing X_n to $K(G_n, n+1)$ yields the checked split-homotopy-tower reconstruction. Taking G_n to be trivial or $\mathbf{C}_2$ is the instance used in the manuscript.

The theorem is parametrized by the family of groups, so no decision procedure for the family is assumed by the type-level proof. The exact Turing-degree lower bound additionally uses the external representation of Kan presentations and the computable group-order readout. Those arguments are proved in Sections 10 and 11.

21.6 Lifts through a Postnikov extension

The `Obstruction` module checks Proposition 11.1. Its first equivalence rewrites a lift with its projection homotopy as a family of points in fibers of the projection. Each fiber of the dependent-sum projection is equivalent to the specified path type. Applying function extensionality also gives an equivalence with a single path from $k \circ u$ to the constant function in the mapping type. The declaration `split_extension` uses any given nullhomotopy to identify the extension with the constant loop family over the base.

The section equivalence takes a comparison $e : \|E\|_m \simeq B$ and a homotopy $e\eta_E \sim p$ as input. The code reindexes the family of fibers defining the section space along e^{-1} and applies e to the fiber paths. The equation $e(e^{-1}(b)) = b$ fixes their endpoints. Finally, univalence transports the section type of the truncation unit along a total-space equivalence. Composing with the two comparison equivalences identifies the nullhomotopy spaces. This is the synthetic step used to distinguish the split and twisted blocks through their total spaces. The nonzero cup-square calculation and the finite Kan readout are the cohomological and representational steps of Section 11.

21.7 Reproducibility and coverage

The supplied check script compiles the proof modules with the HoTT kernel at trust level zero and reports the dependencies of the principal declarations. The verification log records the completed run. The companion declares no extra axioms. The dependency report distinguishes proofs using only identity and tuple laws, proofs using function extensionality, and proofs that use upstream truncation, univalence, or quotient primitives. These are the actual foundations of the cited Lean HoTT library, whose truncation and quotient primitives differ from finite simulation.

The kernel-checked results are listed in Table 1. The manuscript proves the cup-square calculation, simplicial code and oracle-degree theorems, the upper bounds for finite parametrized probes, the operational ledger and strategic results, and the external topos and pro-category comparisons. The source archive contains the manuscript and its figures together with the proof modules and verification log.

22 Scope and conclusions

22.1 Three forms of adequacy

For a proposed network model, semantic adequacy asks whether the witness type contains exactly the intended forms of agreement and their equivalences. The homotopy pullback and its path formulas answer this question relative to the chosen interfaces. A shared receipt namespace can be a set or a higher type. The comparison changes the model precisely when it changes the possible paths. Coherence of whole families can be examined by sections and finite probes. A finite truncation can be used for a specified observation when it meets the proved dimension bound.

Computational adequacy asks whether an agent can produce or evaluate the requested finite presentation. The representation promises matter. A degreewise finite Kan code can answer finite homotopy group queries because it supplies complete lists and the Kan condition. A descriptor that only enumerates increasingly many candidates has a weaker interface. For the explicit ideal objects, exact uniform presentation has the oracle degree of the hidden sequence, even if the ordinary homotopy groups are fixed. Pointwise convergent finite hypotheses have a lower information requirement, and their validity may depend on which direction of evidence is used.

Operational adequacy asks whether an enabled sequence of transitions can actually realize and pay for an agreement. A homotopical gluing theorem supplies a joint witness. A settlement can use the witness only under its authority, balance, time and delivery conditions. Conversely a protocol can make steady, computable progress on provisional reports without eventually knowing a computable threshold for their exactness. The cofinal acquisition theorem states one set of hypotheses under which the necessary resources become available to the agent.

22.2 Semantic boundaries

The non-computability statements concern a declared class of presentations. They are robust under equivalence of spaces within that class, and the enumerable Kan example gives a stronger obstruction when complete lists are weakened to enumeration. A different presentation model can change the reduction from codes to invariants. The examples do not require an internal effective ∞-topos or identify physical supply with a non-computable oracle. They isolate the semantic class that contains both the presented effective towers and ideal towers with the stated lower bounds.

The network semantics uses one directed execution model and one reversible witness calculus. When a directed higher model of execution is supplied, it can

use these witness types in its mapping spaces or over its histories. Establishing that a proposed directed model preserves all operational invariants requires its own adequacy theorem. Similarly, a strategic theorem that compares equilibria of two presentations must keep the allowed deviations and account for the value of information. The results in this paper give those comparison hypotheses explicitly.

The ideal completion preserves the composition law needed by the effective networks. It adds objects whose finite observations remain individually presentable while their total coherent information is beyond any effective listing. The separation is visible in the split tower through group orders and in the extension-class tower through a finite coherent-lift test. Both constructions show how a participant can be fully effective on the finite witness it receives while interacting with an ideal object that no uniformly effective service can present.

22.3 Authorship and proof evidence

This manuscript is prepared with an anonymous byline for editorial review. AI assistance was used in drafting the exposition, constructing the Lean proof terms, and assembling the reproducibility materials. The formal evidence is the source and kernel verification described in Section 21. The analytic evidence consists of the definitions and proofs in the text and the cited results they invoke. The responsible submitting author should review these claims and the journal's terms before submission.

A Additional calculations and reference specifications

A.1 A cycle with and without a coherence condition

Consider a triangle of local components. There is one witness at each vertex and all three mediators are the same pointed type B. Every leg sends its witness to the basepoint. The incidence-diagram match of (5) is then $(\Omega B)^3$, with one loop for each edge. There is no condition in the incidence diagram relating the three seam paths. A protocol may impose a comparison around the cycle. For example it can require that traversing the three changes of verification plan agrees with the fixed plan, with the agreement recorded as an identity witness. The new apex is

$$T(B) = \sum_{p:\Omega B} \sum_{q:\Omega B} \sum_{r:\Omega B} \big((p \cdot q) \cdot r = \mathrm{refl}\big).$$

For each p, q, left multiplication by $p \cdot q$ is an equivalence of ΩB, with inverse left multiplication by the inverse loop. Its homotopy fiber over the reflexive loop is

contractible. Thus projection gives

$$T(B) \simeq (\Omega B)^2.$$

An inverse chooses $r = (p \cdot q)^{-1}$ and the specified inverse-law path. The identity of the round trip uses the contraction of the whole fiber, including its path witness. This proof does not replace the final constraint by a Boolean comparison. In $B = BG$ the formula becomes the familiar equation $pqr = 1$ in a group, with the third factor determined by the first two. In $B = K(A, 2)$ the equation is a type of paths between loops, and the equivalence preserves the higher comparisons between solutions.

If a designer adds two separate proofs that the cyclic composite is reflexive and requires them to be equal, the resulting data can be equivalent to one such proof when the second proof is completely unconstrained except for equality with the first. Indeed the extra choice is a pointed singleton in the path type. If the second proof has an independent interface, comparing the two may be a substantive extra requirement. This explains why the number of written equalities is not alone a valid measure of needed homotopy depth. The relevant bound follows from the shape and free boundary of the compatibility problem.

A.2 Nonuniform snapshots and relative computation

A useful specification for a provider of observations separates its supported query type from its correctness claim. For an ideal bit object X^b, an exact depth provider returns one valid Kan code of the named stage on each request n. A tracked provider returns a code on request (n, s) and promises eventual correctness at each fixed n. A versioned bounded provider only promises that its response reflects a stated finite run. A verifier can use the last promise to settle a computation fee without assuming a limiting observation. The required code and evidence differ in the three cases.

Service	Mathematical claim	Information for the split family
Individual snapshot	For this n, a code of X^b_n is provided.	The finite prefix through n is sufficient.
Exact uniform	One rule supplies exact codes for every n.	The least oracle degree is the degree of b.
Tracked hypotheses	One rule supplies codes on (n, s), eventually exact at each n.	An ordinary effective rule exists exactly for limit-computable b.
Bounded run	The output reflects the declared finite computation.	An ordinary verifier can check the run when supplied with its trace.

For the extension-class family replace a prefix of groups by a prefix of extension choices, and use the finite probes of Proposition 11.3 as readouts. The relative version of the tracked criterion follows by repeating the two reductions of Theorem 12.2 with an oracle D. In one direction the relative algorithm constructs stage codes from pointwise provisional bits. In the other, obtain a relative computable approximation to $b(e)$ by applying the finite lift test to the output code at depth $2m_e - 1$. For all sufficiently large s, that stage has the correct homotopy type. The finite diagnostic works on any valid Kan code of that type. In particular the existence of a D-computable tracked provider is equivalent to $b \leq_T D'$, where D' is the halting jump of D. The usual proof of the limit lemma relativizes because the search for the next change can use D, and D' decides termination of that search.

This argument shows that an intermediary with an exact provider for all depths can answer the bit queries, even if the original input is a package of arbitrary Kan codes. Conversely an intermediary with the bits can produce the full higher observations using the cocycle constructors. The computational work of translating a given bit word into an explicit higher presentation can be large. The oracle equivalence describes the uniform information needed, independently of that work.

A.3 An explicit finite calculation for a stage descriptor

The canonical cocycle constructor can be specified without a simplicial model checker. Given A, m, k, it enumerates all assignments of an element of A to each increasing $(m+1)$-tuple of vertices in $[k]$. For each assignment it forms the alternating sums on increasing $(m+2)$-tuples. It emits the assignment exactly when every sum is zero. An implementation can instead use the bijection of Lemma 10.1, assigning values on the tuples containing zero and extending by a coboundary. Either algorithm produces the same complete list, up to its enumeration order. A simplex map is computed by the monotone vertex map, interpreting degenerate tuples as zero. The proof of the Kan condition is the simplicial-group theorem, which applies uniformly in every degree.

A witness code for the non-split block uses two cochains, as in Lemma 11.2. One can enumerate the first x, compute $x \smile x$, and solve for the second v as a finite linear system over $\mathbb{F}_2$. The coboundary is a linear map in v, so the set of solutions is an affine space when nonempty. Existence on each simplex also follows because a simplex has trivial cohomology in positive degrees. This local solvability is compatible with nonexistence of the specified global lift from Q_m. The obstruction belongs to the coherent family of solutions, rather than the individual system over each simplex.

For the probe test, the domain Q_m is an explicitly presented finite simplicial set. Choose a vertex in a valid target presentation and send the probe's vertex to it. Enumerate assignments for the nondegenerate simplices in order of dimension. When one image is proposed, require its faces to agree with the chosen images of the already assigned lower-dimensional faces, using the degeneracy operations for degenerate faces. An assignment accepted in every degree defines a simplicial map. Then use the elementary relation in degree $m + 1$ to decide whether the image of the distinguished m-simplex is null. Because the target is Kan, the existence of a strict map of the required kind is equivalent to the existence of its pointed homotopy class. The computation is a decision procedure on each supplied code, backed by the general topological proof of what it decides.

A.4 A margin theorem for irreversible choices

Sometimes a controller has a validated interval $[L(a), U(a)]$ for the payoff of every action a in a finite admissible set. Suppose the intervals are simultaneously correct, meaning $L(a) \le u(a) \le U(a)$ for the same unknown payoff function u. If an action a_0 has

$$L(a_0) \;\ge\; \max_{a \neq a_0} U(a),$$

then a_0 is an exact optimizer for every compatible payoff function. For a proof, compare $u(a_0) \ge L(a_0)$ with $U(a) \ge u(a)$ for each competitor. If the lower bound is smaller than the largest competitor upper bound by at most δ, the same argument gives regret at most δ. In a strategic profile one may apply the inequality to a baseline and its individual deviation outcomes when each outcome is admissible and the baseline's payoff is correctly bracketed.

If every action has an interval shrinking with further search and the true maximizer has a strictly positive gap from the competitors, eventually the first inequality holds. Indeed, take any error less than half the gap on the baseline and the competitors, and use the finite maximum of the stage bounds at which all interval endpoints lie within that error. This conclusion depends on the intervals being sound at every stage. Eventual correctness of a hypothesis alone gives no such interval. For the halting example a negative provisional answer can be compatible with both limiting values until an additional certificate or trusted source is supplied. A nontrivial decision margin is therefore evidence only when its input bounds are justified by the verification model.

A.5 A guide to the assumptions used in the main constructions

The structural proofs about paths, composition and coherent families use dependent sums, products and function extensionality. The truncation comparisons

additionally use higher inductive truncation and its path theorem. The explicit split family uses Eilenberg–Mac Lane objects, with an external Kan presentation for the computability claims. The family with fixed groups also uses representability of ordinary cohomology and the mod-two cup product. The pro realization example uses classical choice in extending a linear functional on a quotient vector space, and the existence result about coherent points in finite stages uses the infinite-branch argument. These uses of choice are separate from the constructive equivalences.

The ledger theorems assume a specified authority boundary and guarded atomic effects. Their liveness consequences additionally assume persistent enabling and an appropriate scheduler. The bargaining arguments assume defined utilities and deviation rules. The observation convergence claims assume the access budget grows cofinally in the requested search stages or that a provider actually supplies the required prefix. None of these external assumptions is part of the type of a homotopy pullback. They are the premises that allow a higher witness theorem to be used in a concrete negotiation.

References

[1] I. Barnea, Y. Harpaz, and G. Horel, *Pro-categories in homotopy theory*. Algebraic & Geometric Topology 17 (2017), 567–643. arXiv:1507.01564.

[2] J. C. Baez, K. Courser, and C. Vasilakopoulou, *Structured versus decorated cospans*. Compositionality 4 (2022), article 3. arXiv:2101.09363.

[3] A. K. Bousfield and D. M. Kan, *Homotopy Limits, Completions and Localizations*. Lecture Notes in Mathematics 304. Springer, 1972. Publisher.

[4] V. Brattka, G. Gherardi, and A. Pauly, *Weihrauch complexity in computable analysis*. In *Handbook of Computability and Complexity in Analysis*. Springer, 2021. arXiv:1707.03202.

[5] U. Buchholtz, F. van Doorn, and E. Rijke, *Higher groups in homotopy type theory*. LICS 2018, 205–214. arXiv:1802.04315.

[6] W. Dean, *Recursive functions*. The Stanford Encyclopedia of Philosophy, especially the discussion of the arithmetical hierarchy and the limit lemma. Entry.

[7] F. van Doorn, J. von Raumer, and U. Buchholtz, *Homotopy type theory in Lean*. Interactive Theorem Proving, LNCS 10499, Springer, 2017. arXiv:1704.06781.

[8] G. Friedman, *An elementary illustrated introduction to simplicial sets*. Rocky Mountain Journal of Mathematics 42 (2012), 353–423. arXiv:0809.4221.

[9] N. Ghani, J. Hedges, V. Winschel, and P. Zahn, *Compositional game theory*. LICS 2018, 472–481. arXiv:1603.04641.

[10] N. Ghani, C. Kupke, A. Lambert, and F. Nordvall Forsberg, *A compositional treatment of iterated open games*. Theoretical Computer Science 741 (2018), 48–57. arXiv:1711.07968.

[11] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002. Section 4.3 on Eilenberg–Mac Lane spaces, cohomology, and obstruction theory. Author's text.

[12] A. Hatcher, *Algebraic Topology*. Cambridge University Press, 2002. Section 3.2 on cup products and real projective space. Author's text.

[13] R. Haugseng, *Iterated spans and classical topological field theories*. Mathematische Zeitschrift 289 (2018), 1427–1488. arXiv:1409.0837.

[14] R. Haugseng, F. Hebestreit, S. Linskens, and J. Nuiten, *Two-variable fibrations, factorisation systems and ∞-categories of spans*. arXiv:2011.11042.

[15] J. M. E. Hyland, *The effective topos*. In A. S. Troelstra and D. van Dalen, eds., *The L. E. J. Brouwer Centenary Symposium*, 165–216. North-Holland, 1982. Author's copy.

[16] N. R. Jennings, P. Faratin, A. R. Lomuscio, S. Parsons, C. Sierra, and M. Wooldridge, *Automated negotiation: prospects, methods and challenges*. Group Decision and Negotiation 10 (2001), 199–215. Bibliographic record and full text.

[17] C. Kapulkin and P. L. Lumsdaine, *The simplicial model of univalent foundations (after Voevodsky)*. Journal of the European Mathematical Society 23 (2021), 2071–2126. Publisher.

[18] L. Lamport, *Time, clocks, and the ordering of events in a distributed system*. Communications of the ACM 21 (1978), 558–565. Author's copy.

[19] Lean developers, *Lean 2 HoTT library*. Source snapshot at commit `8072fdf`.

[20] J. Lurie, *Kerodon*. Theorem 2.5.6.1 and Construction 2.5.6.3. Dold–Kan correspondence. Eilenberg–Mac Lane functor.

[21] J. Lurie, *Kerodon*. Section 3.2.2, The homotopy groups of a Kan complex. Section.

[22] J. Lurie, *Kerodon*. Section 3.5.8, The Postnikov tower of a Kan complex. Section.

[23] J. Lurie, *Higher Topos Theory*. Annals of Mathematics Studies 170. Princeton University Press, 2009. Author's text.

[24] J. Lurie, *Spectral Algebraic Geometry*. Manuscript, 2018, Appendix A.7. Author's text.

[25] N. A. Lynch and M. R. Tuttle, *An introduction to input/output automata*. CWI Quarterly 2 (1989), 219–246. Author's copy.

[26] J. F. Nash, Jr., *The bargaining problem*. Econometrica 18 (1950), 155–162. JSTOR record.

[27] G. Quick, *Profinite homotopy theory*. Documenta Mathematica 13 (2008), 585–612. arXiv:0803.4082.

[28] E. Riehl and M. Shulman, *A type theory for synthetic* $(\infty, 1)$*-categories*. Higher Structures 1 (2017), 147–224. arXiv:1705.07442.

[29] E. Rijke, M. Shulman, and B. Spitters, *Modalities in homotopy type theory*. Logical Methods in Computer Science 16 (2020), article 2. Publisher and full text.

[30] A. Rubinstein, *Perfect equilibrium in a bargaining model*. Econometrica 50 (1982), 97–109. JSTOR record.

[31] A. M. Turing, *On computable numbers, with an application to the Entscheidungsproblem*. Proceedings of the London Mathematical Society, second series 42 (1936–1937), 230–265. Full text.

[32] The Univalent Foundations Program, *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013. Book and source.

[33] M. Wooldridge and N. R. Jennings, *Intelligent agents: theory and practice*. The Knowledge Engineering Review 10 (1995), 115–152. DOI. Author's copy.