Self-Improving Intelligence

A visual guide to representation, discovery, and recursive revision

---

What improves when a system becomes **better at making itself better?**

01	**Repair a case**	a diagnostic policy
02	**Improve discovery**	a better compiler
03	**Acquire a search rule**	a composed predicate
04	**Improve rule acquisition**	a checked search representation

Each step retains its assumptions, evidence, and costs.

Komorebi P.

Guide v1.0.0

Companion to the 60-page first edition

8 October 2026

Analytical constructions, not agent-performance experiments.
The diagrams explain the manuscript; they do not replace its proofs.

ORIENTATION

The route through this guide

Read the worked world first. Then follow the discovery procedure outward, keeping the supplied machinery in view.

03-12
The repair problem
States, observations, trees, uncertainty, and transport.

13-21
Changing the compiler
Certified shortcuts, learned pruning, and repair scope.

22-28
Changing acquisition
Checked equations, preserved evidence, invalidation, and costs.

29-36
What the result supports
Evaluation, continuation, the learned bridge, and source routes.

HOW TO USE IT

Every plate identifies its source section. Manuscript page numbers are the printed numbers, not the PDF viewer index.

WHAT A DIAGRAM MEANS

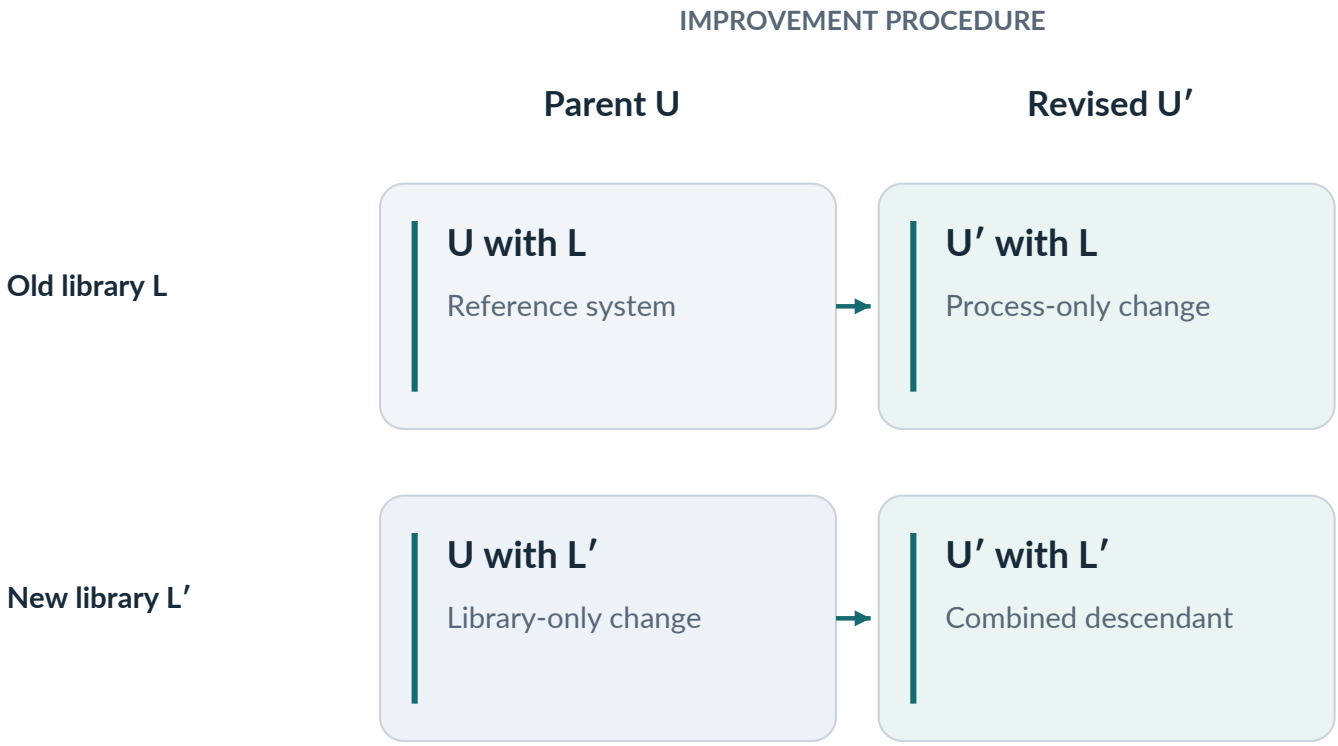
A solid arrow is a specified transition or dependency. Dashed paths are explicitly labeled as recurrence or a qualified extension. Counts retain their original units.

[S] Scope guide; §§1-3 | manuscript pp. iii, 1-4

I / WHAT CHANGES

A better system is not always a better improver

Changing a library and changing the procedure that uses it answer different questions. A crossed comparison keeps them separate.



Counterexample

Storing all four task answers improves retrieval over storing one. The retrieval procedure can remain exactly the same.

READ THE HORIZONTAL COMPARISON

Holding the library fixed isolates a change in the improvement procedure. The combined descendant may improve for a different reason.

NO FREE STARTING ADVANTAGE

Use comparable starting solvers, task laws, installed information, and budgets. A stronger base model or an extra library is another changed component.

[S] §3; Counterexample 3.2 | manuscript pp. 3-4

II / THE DIAGNOSTIC WORLD

One defect. Eight possible cells.

The procedure cannot see the defect index. It receives only the reports of the questions it is allowed to ask.

Possible hidden defect j

0	1	2	3	4	5	6	7

A query names a subset

Ask whether j belongs to A.
The report is 0 or 1.
A query does not move the defect.

A repair names one cell

Repair r_i succeeds only at cell i.
Commit only when every remaining possibility supports it.

The old state is too coarse

“There is one defect” is true in all eight cases. It gives no basis for choosing which cell to repair.

THE LEARNING TARGET

Retain an executable way to find a justified repair in a later episode. Do not install the next episode’s hidden answer.

WHAT IS SUPPLIED

The binary sensors and repair meanings come from a declared finite model. The 88 query-table entries are supplied, not learned through executed calibration.

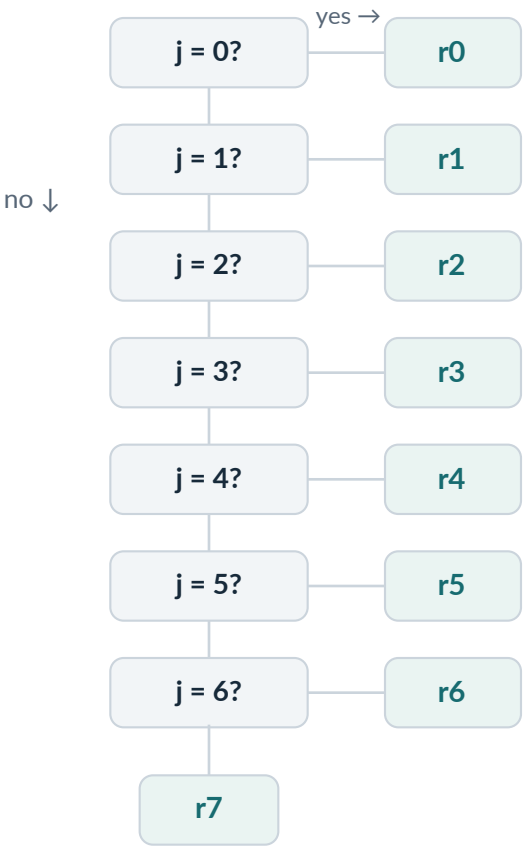
[S] §4; §32 | manuscript pp. 4, 49-51

II / THE DIAGNOSTIC WORLD

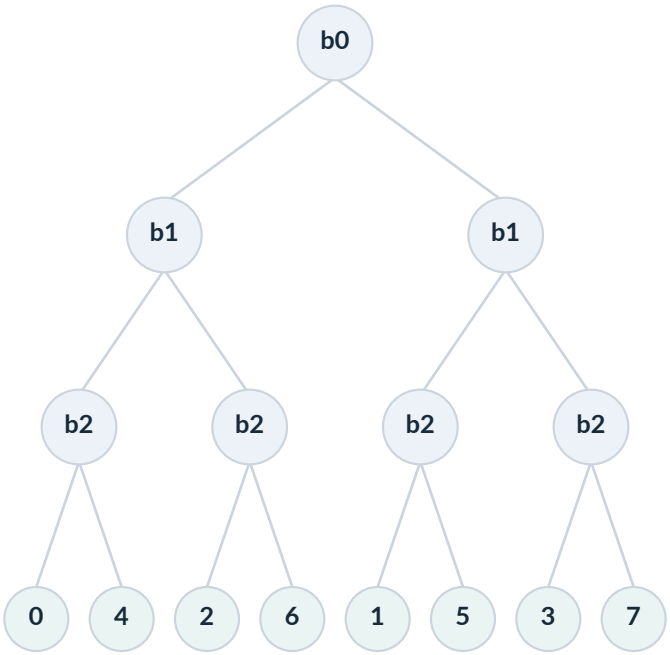
Two investigations, the same installed sensors

Equality-first testing and coordinate queries use the same declared language. Their paths have different depths, but their complete trees have the same size.

Equality-first



Coordinate queries



Leaf number = cell to repair

b0: {1,3,5,7} · b1: {2,3,6,7}
b2: {4,5,6,7}. Left edge: 0. Right: 1.

WORST-CASE EXECUTION

Equality-first: seven questions, then repair. Coordinate queries: three questions, then repair. Total costs are eight and four calls.

COMPLETE STORED CONTROLLER

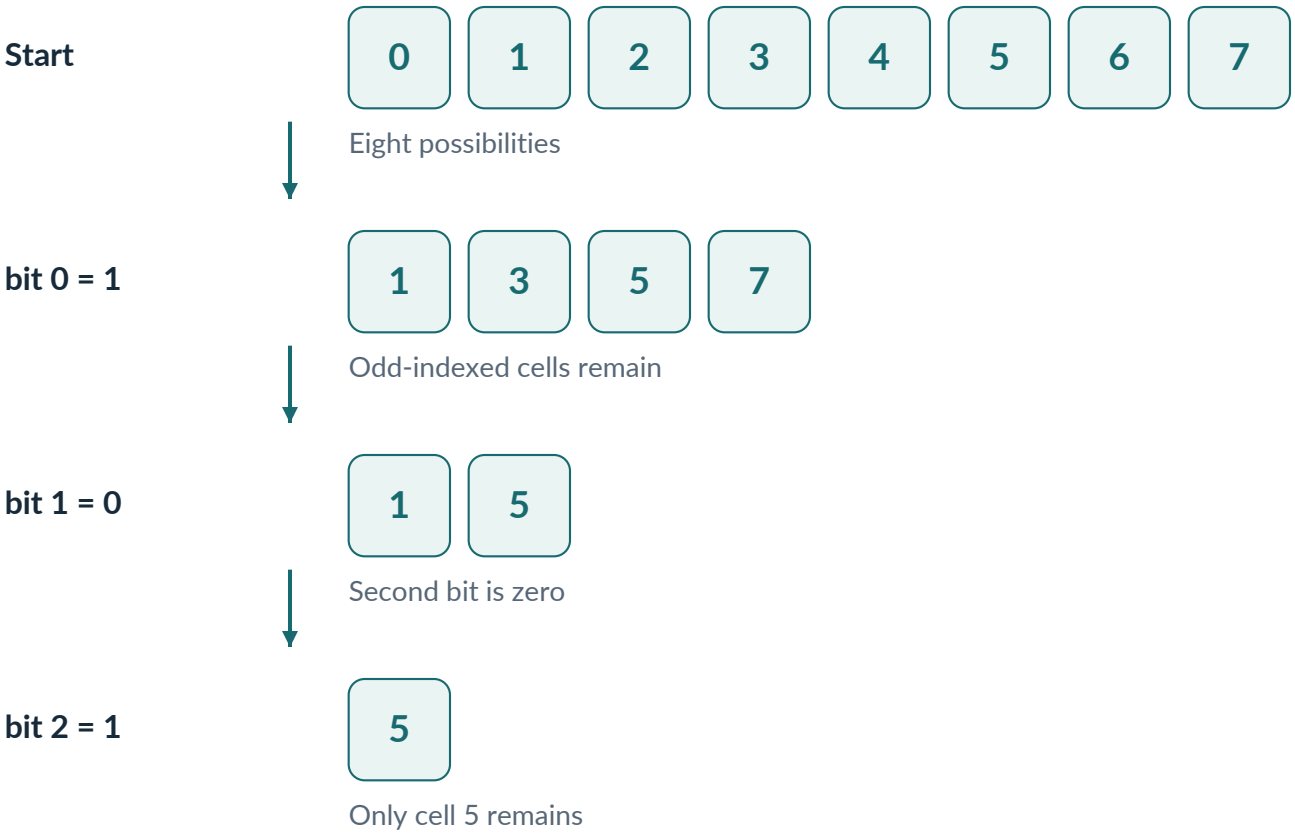
Both full binary trees have eight repair leaves and seven query nodes: fifteen records. These are literal tree records, not bytes or minimal program sizes.

[S] §4; Theorem 32.1 | manuscript pp. 4, 49-50

II / THE DIAGNOSTIC WORLD

Follow one episode: the defect is at cell 5

This page follows the canonical coordinate policy. The procedure knows the reports, not the hidden index shown to the reader.



Execute r5: the fourth call

WHAT WAS ACQUIRED

The three reports select a repair for this episode. The reusable artifact is the branching procedure that works for every allowed starting cell.

ONE EXACT TRACE, NOT THE ONLY TREE

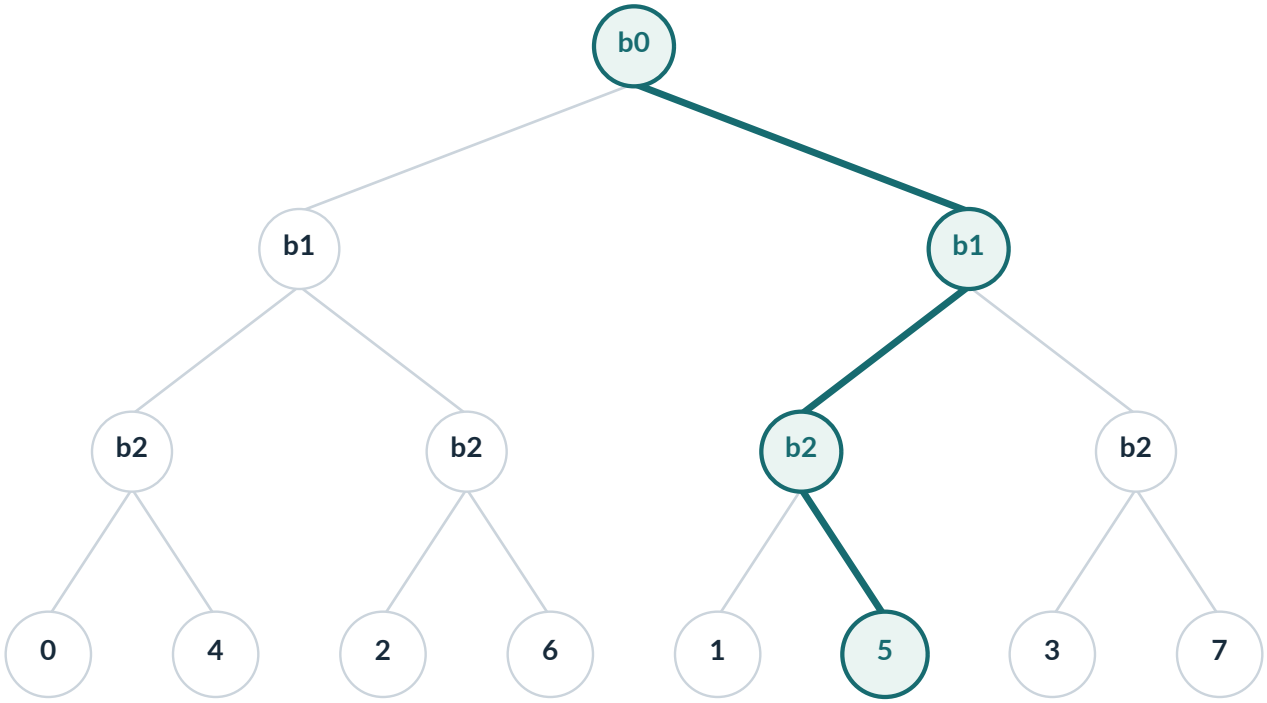
The shipped compiler can use an equality query at the last two-case branch. It has the same cost. This plate deliberately shows the canonical all-coordinate tree.

[S] §4; §32; Appendix A | manuscript pp. 4, 49-54

II / THE DIAGNOSTIC WORLD

The path is not the stored program

A single successful run touches four records. Its complete reusable controller still contains all fifteen.



Report order: bit 0, bit 1, bit 2

4 visited records

15 stored records

COUNT THE TWO RESOURCES SEPARATELY

Execution follows one branch. Literal storage sums every branch. The highlighted path for reports 1, 0, 1 reaches repair 5.

SCOPE OF THE BOUND

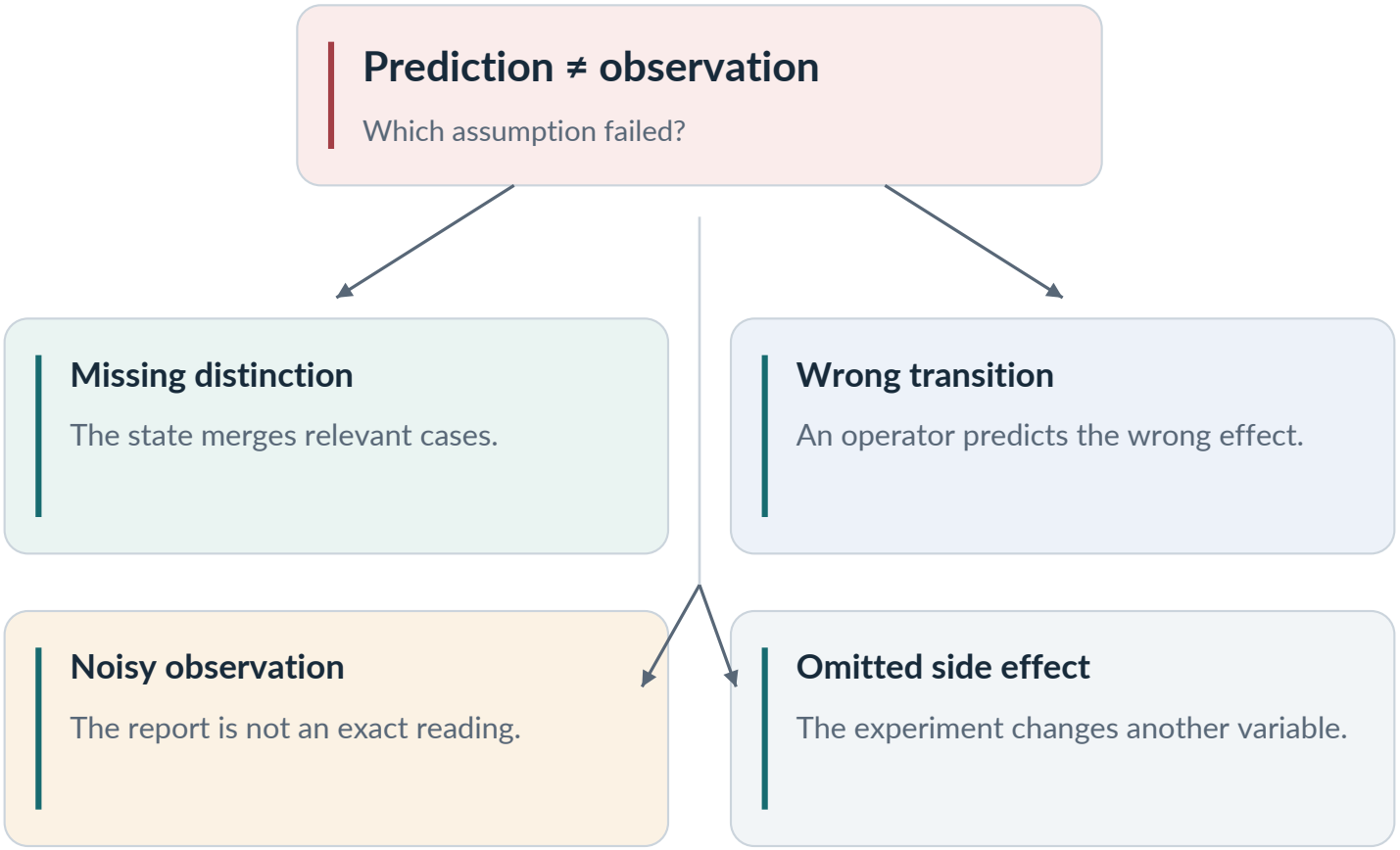
Immutable cases, binary observations, a different successful repair for each case, and one record per node. Shared code or state-reset operations change the model.

[S] §6; Theorem 11.1; §28 | manuscript pp. 6-7, 14-15, 44

II / INVESTIGATION

A residual is a question, not a diagnosis

The same mismatch can be consistent with several causes. A useful next experiment separates repair-relevant explanations.



USE THE FAILURE

Retain competing explanations. Seek an admitted observation whose possible outcomes change which correction is justified.

DO NOT INVENT MISSING ACCESS

A representation cannot recover a distinction unavailable in its complete transcript. With disjoint repairs and identical transcript laws, equal-prior success is at most one half.

[S] §5; Proposition 5.1 | manuscript pp. 5-6

II / INVESTIGATION

The best question need not identify the most

A repair may work for several hidden cases. The right stopping condition is a common successful action, not complete knowledge of the state.

Repair A works on {0,1,2}

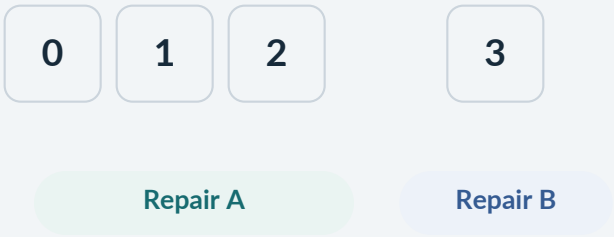
Repair B works on {3}

Balanced 2 / 2 split



Another query is needed on the {2,3} branch.

Repair-aligned 3 / 1 split



Both branches already support one valid action.

BALANCED IS NOT AUTOMATICALLY BEST

The 2/2 question has more information about four equally likely cases, but one branch still mixes incompatible repairs.

THE EXACT OBJECTIVE

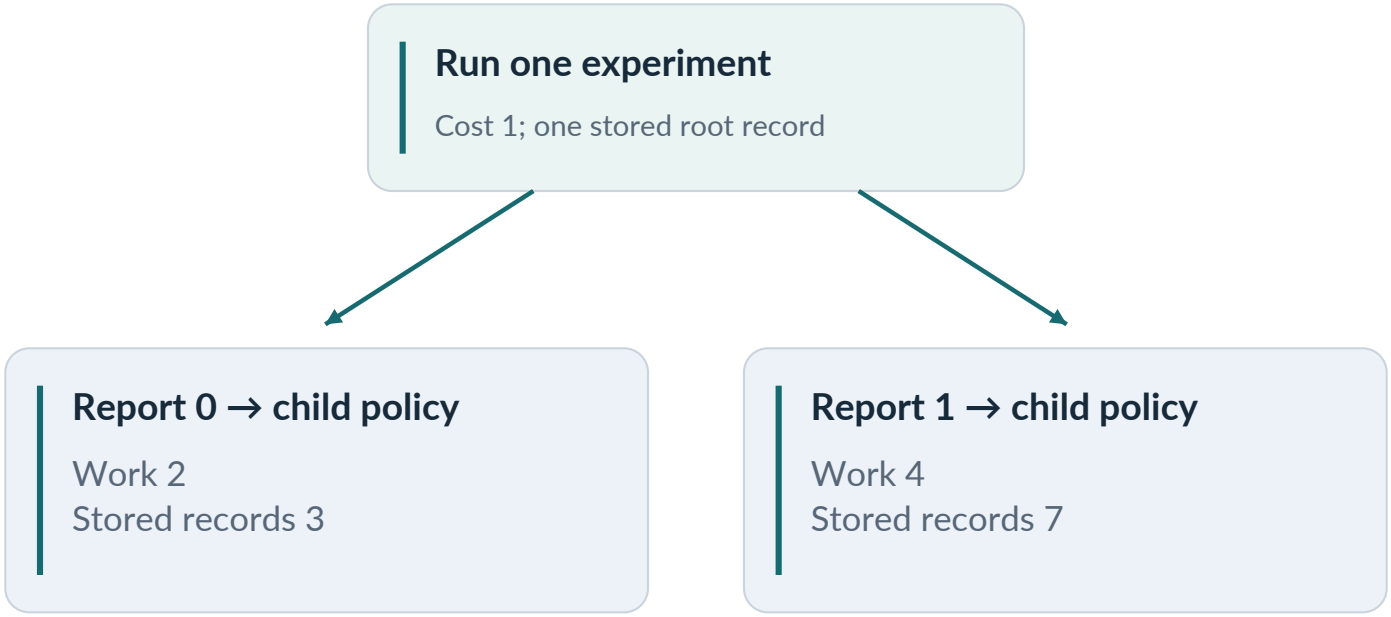
Both questions cost one unit and preserve the state. The 3/1 question supports a correct repair after one observation in every case.

[S] Counterexample 6.3; §5.1 | manuscript pp. 6-8

II / INVESTIGATION

One tree must satisfy both resource limits

The synthesis combines complete child policies. Worst-case execution takes a maximum. Stored controller size takes a sum.



$$\text{Work} = 1 + \max(2, 4) = 5$$

$$\text{Storage} = 1 + 3 + 7 = 11$$

READ THE ARITHMETIC

This illustrative root has cost 1. Its children cost (2 calls, 3 records) and (4 calls, 7 records). The combined tree costs (5 calls, 11 records).

WHAT IS NOT BEING CLAIMED

This is an illustration of the recurrence, not a new optimality result. A real child must be valid on the actual post-experiment support, and every possible branch needs a witness.

[S] Theorem 6.2 | manuscript pp. 6-7

II / INVESTIGATION

An unknown sensor needs a reference

A correct procedure for each known model need not give a common procedure when the model itself is unknown.

Hidden repair bit x; fixed sensor convention θ

Without calibration

$$o = x \oplus \theta$$

Report 0

(0,0) or (1,1)
No common repair.

Reference → θ

Cost 2

Ordinary report → o

Recover x = o ⊕ θ; cost 1

Execute the justified repair

Cost 1

The model tag does not change between steps.

ORDINARY READING ALONE

The report o = x ⊕ θ cannot distinguish (x,θ) = (0,0) from (1,1). They require different repairs. Repeating that same reading adds no information.

THE ADDITIONAL CHANNEL IS PAID

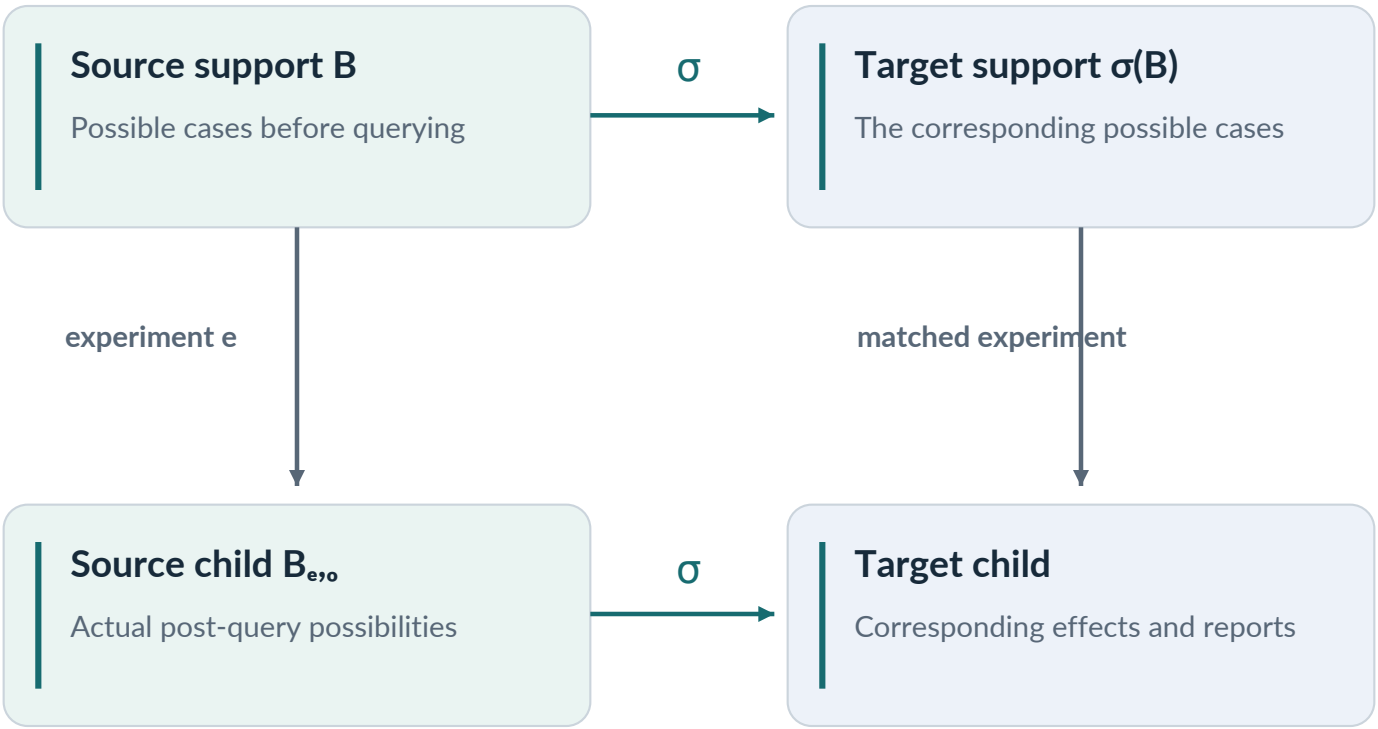
A reference reports θ at cost 2. An ordinary query costs 1, then repair costs 1. Total work is 4, and the complete binary controller stores 7 records.

[S] Theorem 7.1; Counterexample 7.2 | manuscript pp. 8-10

II / TRANSFER

Transport the investigation, not only the repair

A translated answer can be correct while the observation needed to obtain it is unavailable in the target system.



Both routes reach the same target child

A VALID TRANSPORT RETAINS

Experiment permissions, report labels, state effects, successful-repair meanings, and costs. Correspondences are supplied in this finite theorem.

THE MISSING-SENSOR OBSTRUCTION

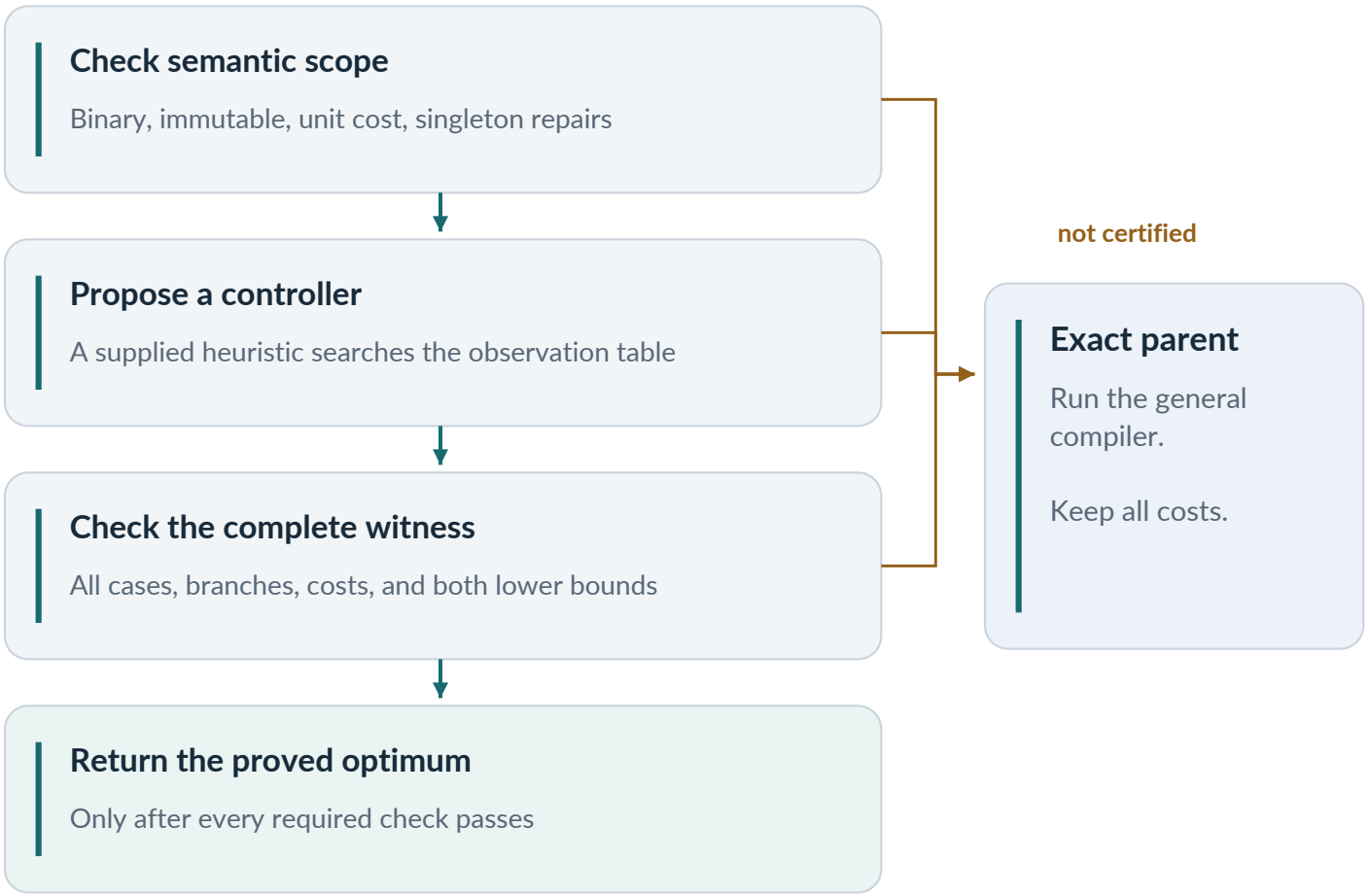
Copying the two possible repairs into a world with no bit-revealing observation does not transport the policy that chooses between them.

[S] Theorem 9.1; Counterexample 9.2 | manuscript pp. 12-13

III / IMPROVING DISCOVERY

Now improve the procedure that builds the tree

A checked candidate can end an exhaustive search only when it is valid and attains the relevant global lower bounds.



THE DECISIVE CERTIFICATE

For n immutable cases with binary unit-cost queries and singleton repairs, a valid tree attaining both 1 + ceil($\log_2$ n) calls and 2n − 1 records closes the frontier.

A FAILED PROPOSAL IS NOT INFEASIBILITY

An invalid scope, missing candidate, or unattained bound sends control to the exact parent. That preliminary work still counts.

[S] Theorem 11.1; §11.2 | manuscript pp. 14-15

III / IMPROVING DISCOVERY

What the shortcut actually reads

On the sixteen-case model, both compilers return an optimal controller with 5 execution calls and 31 stored records.

576,128

exact parent: model-record reads

1,728

shortcut: model-record reads

Scope guard	320
Construct the proposed tree	1,280
Reconstruct branch supports	64
Replay every case	64

320 + 1,280 + 64 + 64 = 1,728

A PRECISE SAVING

The recorded compilation count is 576,128 reads for the parent and 1,728 for the shortcut. Both include completed witness checking.

KEEP THE BOUNDARY VISIBLE

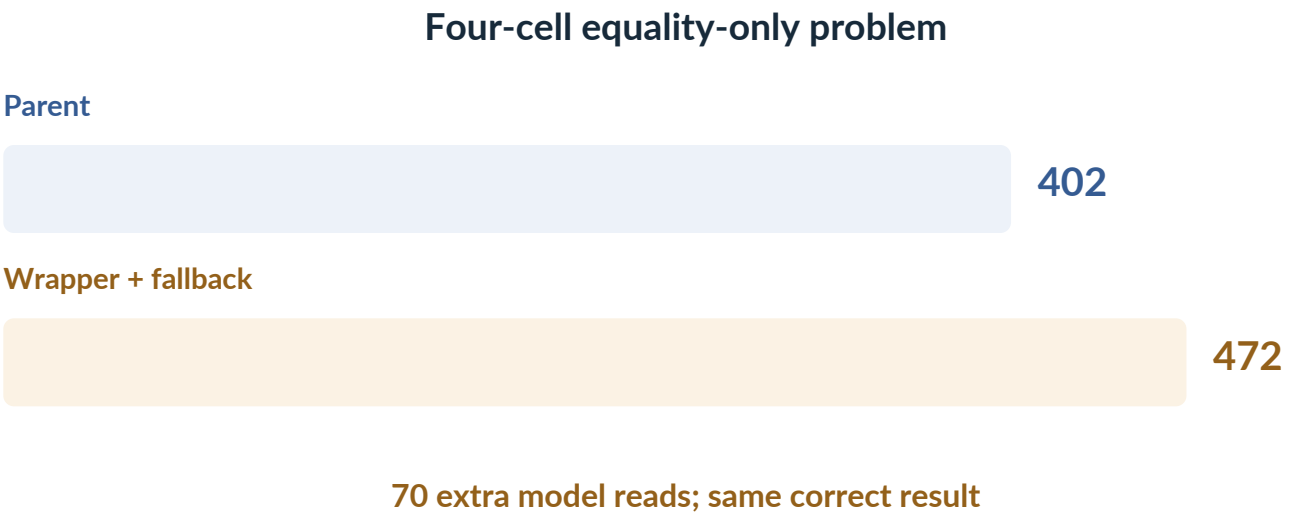
Each also pays 320 common input-validation reads. Selection among all candidates costs 21,124 development reads. Table accesses are not total CPU instructions.

[S] Proposition 12.1; §13.3 | manuscript pp. 16, 19

III / IMPROVING DISCOVERY

A fallback preserves meaning, not free speed

A preliminary shortcut can fail to prove an optimum and then spend extra work before invoking the same exact parent.



SUCCESS

A checked tree fits the contract.

INFEASIBLE

Complete search rules out this finite target.

INCOMPLETE

The allowance ended before a conclusion.

THE EQUALITY-ONLY CONTROL

With four cells and only singleton queries, the shortcut falls back. Both return the same optimum. The wrapper uses 472 reads instead of 402.

KEEP THESE OUTCOMES DIFFERENT

A successful witness, a proved finite-horizon impossibility, and an interrupted search are different results. A timeout cannot certify infeasibility.

[S] Counterexample 13.1; §16.2 | manuscript pp. 18, 24-25

IV / LEARNING A SEARCH RULE

Sometimes the search state is too coarse

No longer formula can recover a feature the learner was never given. Here, the missing distinction is remaining branch capacity.

Same visible prefix: a = 0, b = 3, m = 4

Child capacity C = 2

Reject this root

Child capacity C = 4

Capacity still possible

THE CONFLICT IS EXACT

Both contexts have (a,b,m) = (0,3,4). With capacity 2 the root must fail. With capacity 4, a completion with branch sizes (1,3) is possible.

WHAT THE LEARNER DID NOT INVENT

Capacity is exposed from a supplied feature registry. This repairs an internal search representation, not a physical sensor model or the definition of depth.

[S] Proposition 14.1; Counterexample 14.2 | manuscript pp. 20-21

IV / LEARNING A SEARCH RULE

Four proposals, three counterexamples

The learner composes an arithmetic predicate. Retained counterexamples prevent it from proposing the same rejected behavior again.

1. **a > a**
   (a,b,m,C) = (0,2,2,1)
   Misses an overfull branch.

2. **b > a**
   (0,1,2,1)
   Rejects a feasible completion.

3. **b > C**
   (2,0,2,1)
   Misses too many zeros.

4. **max(a,b) > C**
   All 198 development contexts
   Accepted on the finite domain.

THE ACQUIRED OBJECT

The final five-node expression is max(a,b) > C. The reference learner visits 190 of 3,960 candidates and makes four teacher calls.

FINITE SUCCESS HAS A SCOPE

The teacher checks 198 developmental contexts. That certifies this finite agreement, not validity at every larger capacity. Installation needs a separate scope argument.

[S] Theorem 15.1; §15.2; rule trace | manuscript pp. 21-23

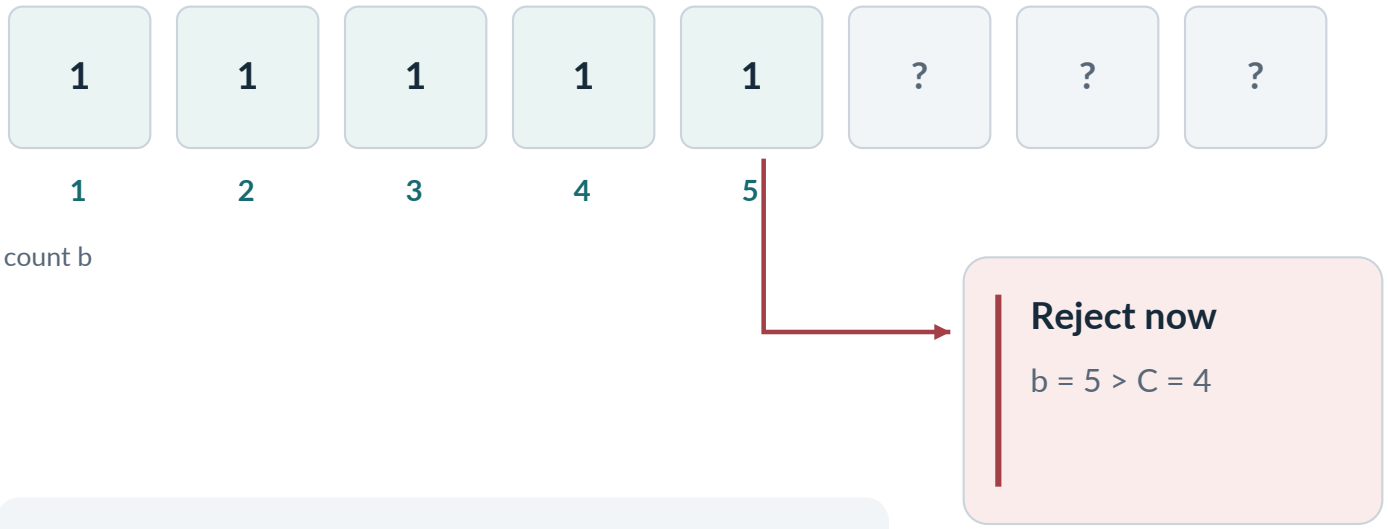
IV / LEARNING A SEARCH RULE

Stop reading when no suffix can rescue the root

Eight cases must fit into two branches of capacity four. After five equal reports, the outcome is already decided.

m = 8 remaining query depth d = 3 C = 4

Read this query's model rows in their fixed order



Three rows stay unread

Whatever they say, the branch already contains too many cases.

WHY PRUNING IS SOUND

Counts cannot decrease. Once one count exceeds C, no unread report can bring that branch back within capacity. The suffix cannot change rejection.

NECESSARY IS NOT SUFFICIENT

Passing the capacity test does not prove that the remaining query library can separate either child. The compiler must still search and verify complete subtrees.

[S] Proposition 16.1; Theorem 16.3 | manuscript pp. 23-25

IV / LEARNING A SEARCH RULE

A training-perfect rule can prune the answer

A dangerous shortcut is not made safe by a final answer checker: it may delete the only successful candidate before that checker sees it.

Development capacities: 1, 2, 4

These two formulas agree throughout the development domain.

Supported rule

max(a,b) > C

5 > 8 is false

Keep investigating.
No capacity obstruction yet.

Training-perfect impostor

max(a,b) > min(C,4)

5 > 4 is true

Incorrectly discard
a feasible completion.

THE LARGER CONTEXT

At (a,b,m,C) = (0,5,16,8), the bounded-cap formula rejects. But the unread cases can complete both branches to eight, so capacity remains feasible.

SEPARATE SELECTION FROM ADMISSION

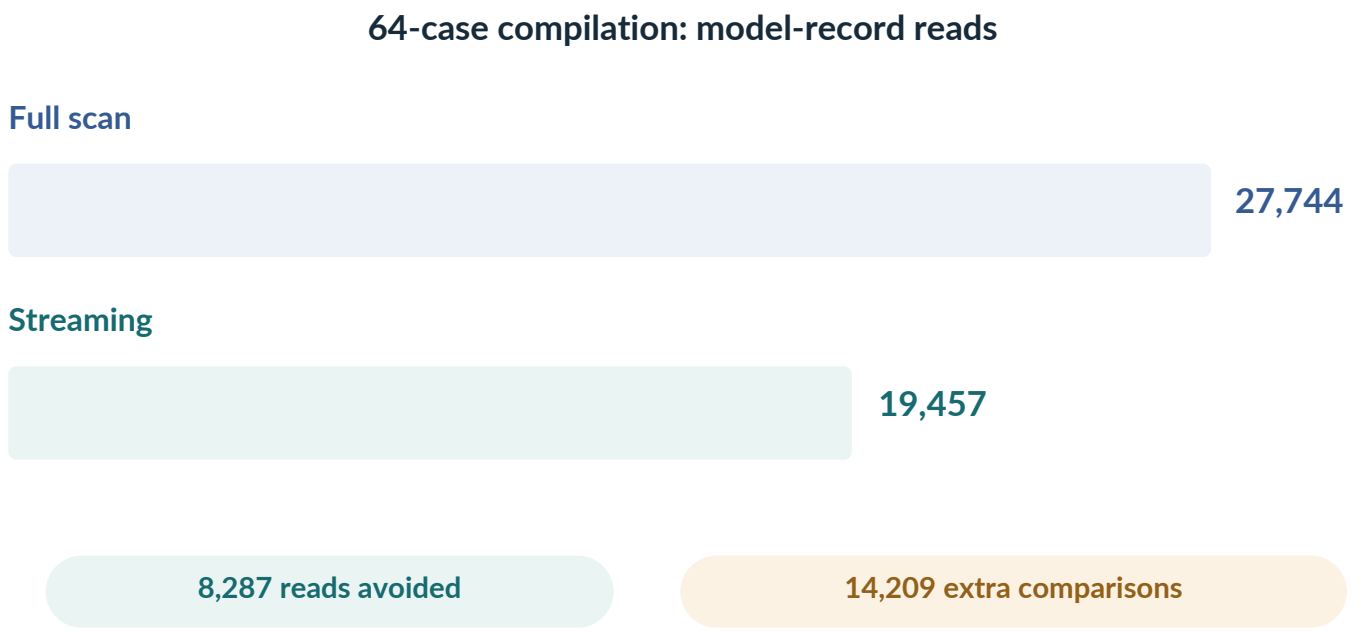
Examples help select a formula. A proof with applicable premises authorizes deleting possibilities. The supplied structural gate accepts only the supported rule form.

[S] Counterexample 16.2 | manuscript pp. 23-24

IV / LEARNING A SEARCH RULE

Fewer model reads can still mean more work

The same ordered compiler returns the same literal controller. Streaming changes when impossible roots are abandoned.



Declare a common price before subtracting

If a read costs w and a comparison costs 1, saving = 8,287w − 14,209. It is positive only when w > 14,209 / 8,287.

WHAT THE COUNT SUPPORTS

At 64 cases, streaming avoids 8,287 model rows. The returned controller still costs 7 online calls and 127 stored records.

NO UNIVERSAL SPEEDUP

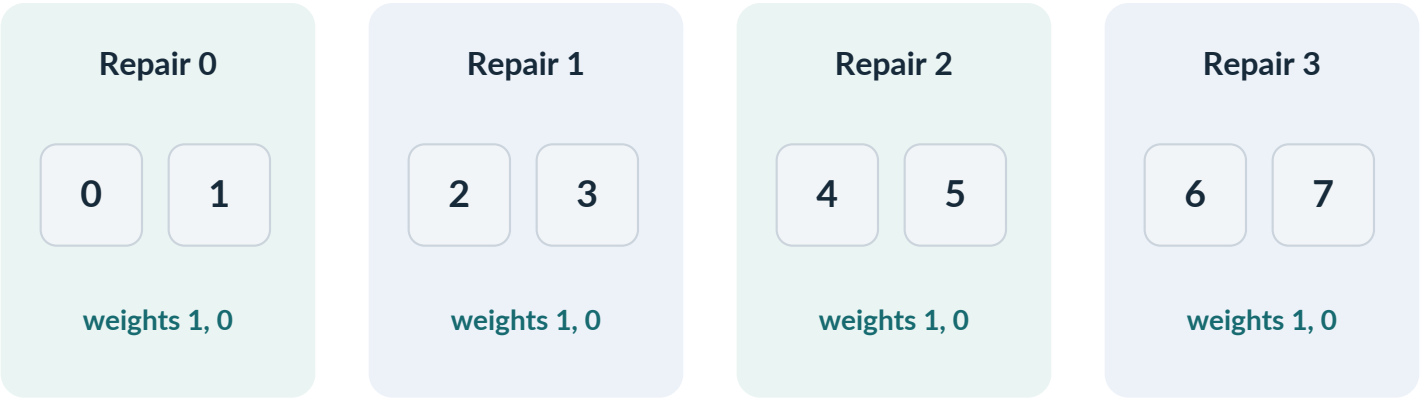
Streaming also adds 14,209 comparisons in the conservative account. At equal prices for reads and comparisons, that isolated modification costs more.

[S] §18; Theorem 16.3 | manuscript pp. 24-28

V / EXTENDING THE REPAIR SCOPE

Count repair distinctions, not hidden states

If one repair fixes a pair of cases, a diagnostic policy need not identify which member of the pair occurred.



8 hidden cases → 4 repair groups

Total weight 4 requires at least 4 leaves.
The supplied pair-index queries attain depth 2.

2 questions + 1 repair | 7 stored records

THE WEIGHT CERTIFICATE

Put nonnegative weights on cases so every repair-success set has total at most 1. A successful leaf then carries at most 1 unit. Binary depth h supports at most 2^h leaves.

A CHANGED REPAIR CAN BREAK THE BOUND

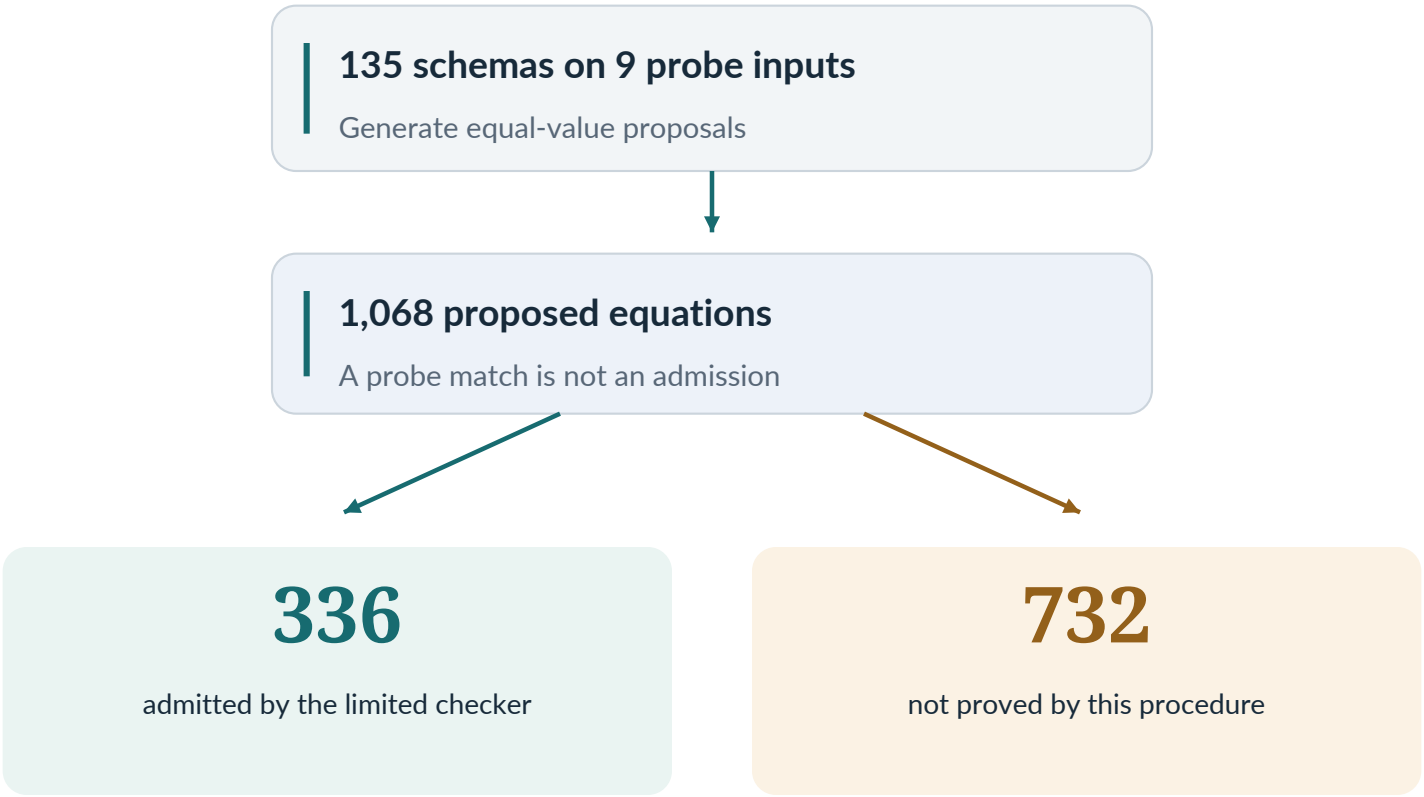
Adding a universal repair makes the old total weight of 4 violate its success-set constraint. Recheck the weights before using the old pruning rule.

[S] Propositions 17.1, 22.1; Theorem 22.2 | manuscript pp. 25, 33-35

VI / IMPROVING ACQUISITION

A useful equation begins as a proposal

The next acquired object is a task-independent library of equations. Evaluations suggest candidates; derivations decide which can be used.



Example: max(x,1) = x matches x = 1,2,3.
It fails at x = 0 and is not admitted.

WHAT IS ACQUIRED

The finite proposer constructs 1,068 equations from equal probe vectors. The limited proof procedure admits 336 with replayable arithmetic derivations.

UNPROVED IS NOT FALSE

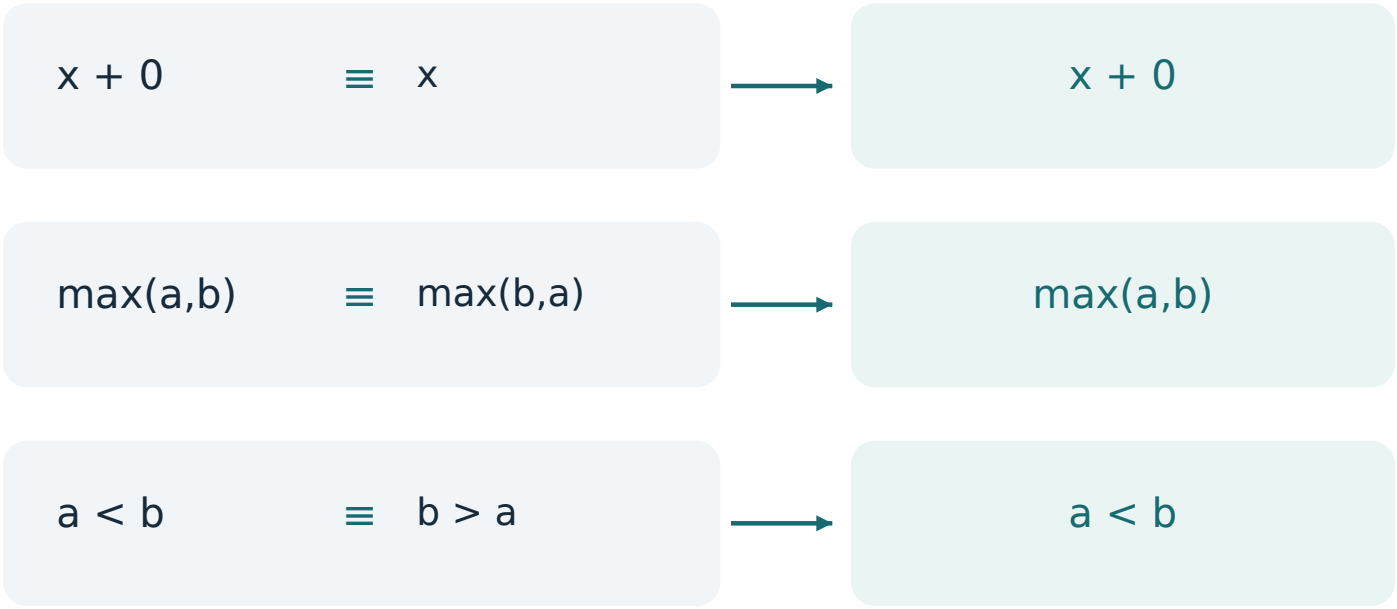
The remaining 732 may be false, valid only on a narrower scope, or simply beyond this proof search. The primitive arithmetic axioms and checking machinery are supplied.

VI / IMPROVING ACQUISITION

Keep the earliest checked representative

Equivalent expressions can share a search representative without changing the original program submitted to the teacher.

Schematic classes; left entry is assumed earlier



3,960 expressions → 1,362 representatives

Stronger supplied normalizer: 1,352 representatives.
The acquired library is not claimed to beat that prior.

WHAT IS PRESERVED

The exact proposed program trees, retained counterexamples, final result, and exhaustion outcome. Removed later equivalents would already fail a retained counterexample.

WHAT THE THEOREM NEEDS

Pure total arithmetic, sound equations, fixed order, an extensional deterministic teacher, and no forgotten counterexamples. It does not preserve every possible cost-sensitive objective.

[S] Theorems 21.1-21.2 | manuscript pp. 30-32

VI / IMPROVING ACQUISITION

Less repetition. The same corrective evidence.

The quotient changes which redundant candidates are evaluated. It does not change the task teacher or smuggle in the target rule.

Original enumeration	Acquired quotient
190 candidates visited on the capacity task	**76** candidates visited on the same task
1,898 interpreter-node visits	**1,376** interpreter-node visits

The evidence transcript does not shrink

a > a → b > a → b > C → max(a,b) > C
Same proposals; same counterexamples; same accepted program.

COLD REACQUISITION

Both learners begin the new task with no task-specific counterexamples. They submit the same four programs and receive the same three counterexamples.

COSTS STILL OUTSIDE THESE NUMBERS

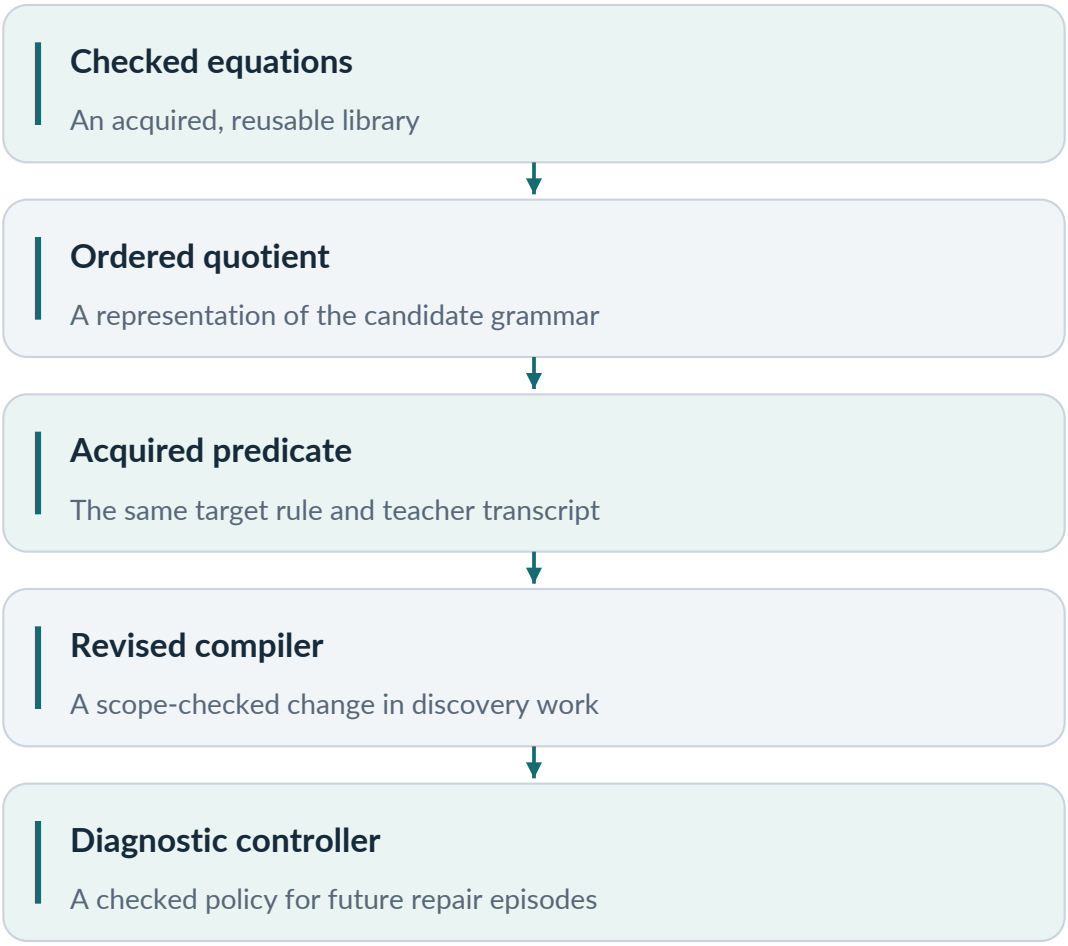
Equation discovery, proof admission, quotient preparation, dispatch, and storage must be charged separately. The ten targets are finite tasks in one arithmetic language.

[S] §21.1; Theorem 21.2 | manuscript pp. 30-32

VI / IMPROVING ACQUISITION

One chain, not multiplied headline gains

The representation that speeds up rule acquisition feeds a separately certified compiler change. Each link keeps its own comparison.



THE COMPOSITE CLAIM

If the quotient preserves rule acquisition and the acquired rule passes the unchanged compiler gate, the complete chain returns the same rule and controller.

MEANING AND SAVINGS ARE SEPARATE

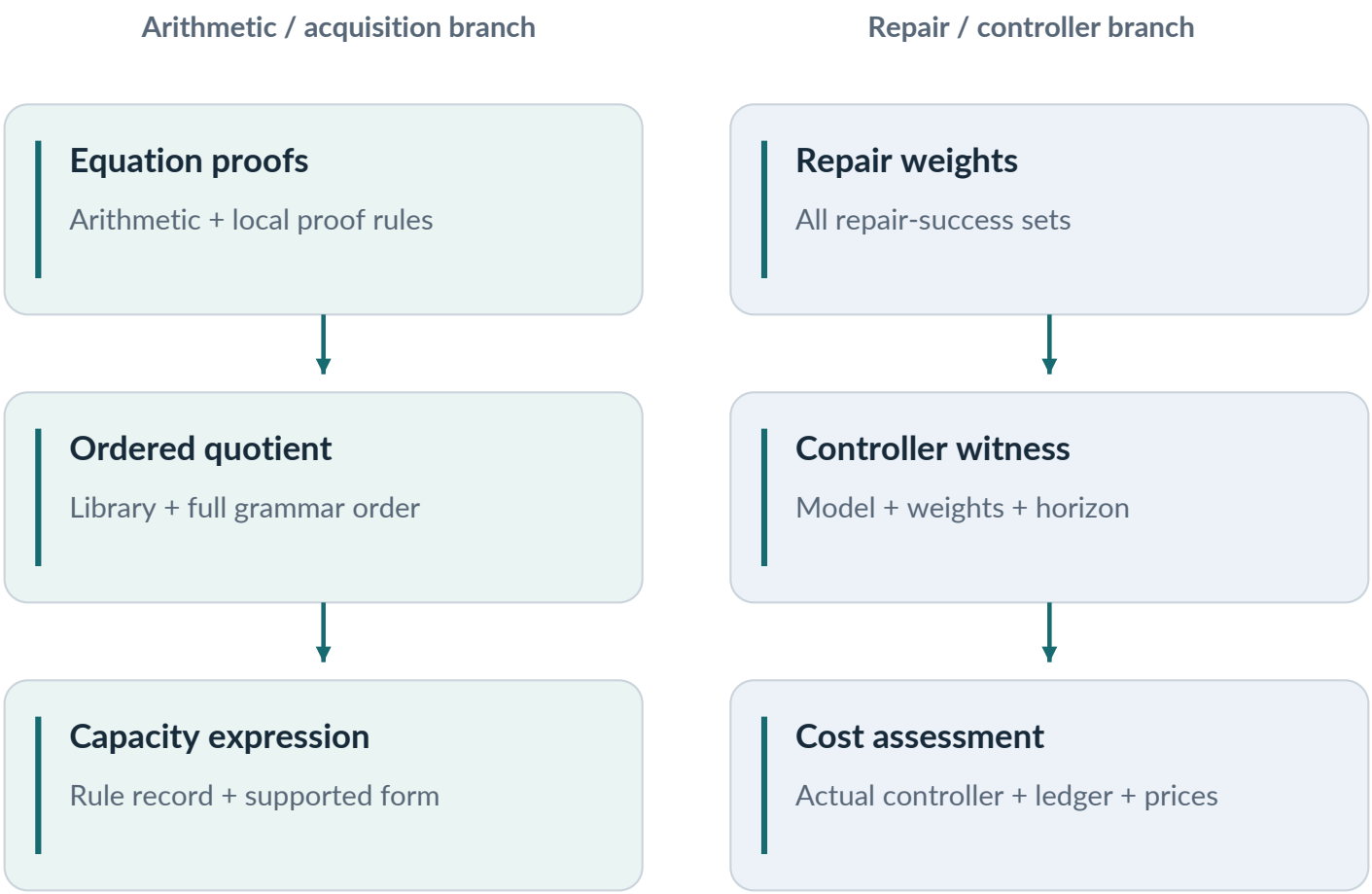
Let K be extra preparation. The composite saves work in a common additive unit only if $(A - A') + (D - D') > K$. Do not multiply ratios from different baselines or units.

[S] Proposition 23.1; §25 | manuscript pp. 35-40

VII / CONTINUED APPLICABILITY

A certificate has a dependency address

When a premise changes, mark the affected claims and their descendants. Reuse unaffected witnesses only under a complete dependency declaration.



The branches are not interchangeable certificates.

SELECTIVE RECHECKING

Arithmetic equations and a repair-weight witness belong to different branches. A change in one branch need not invalidate everything.

THE GRAPH IS A PREMISE

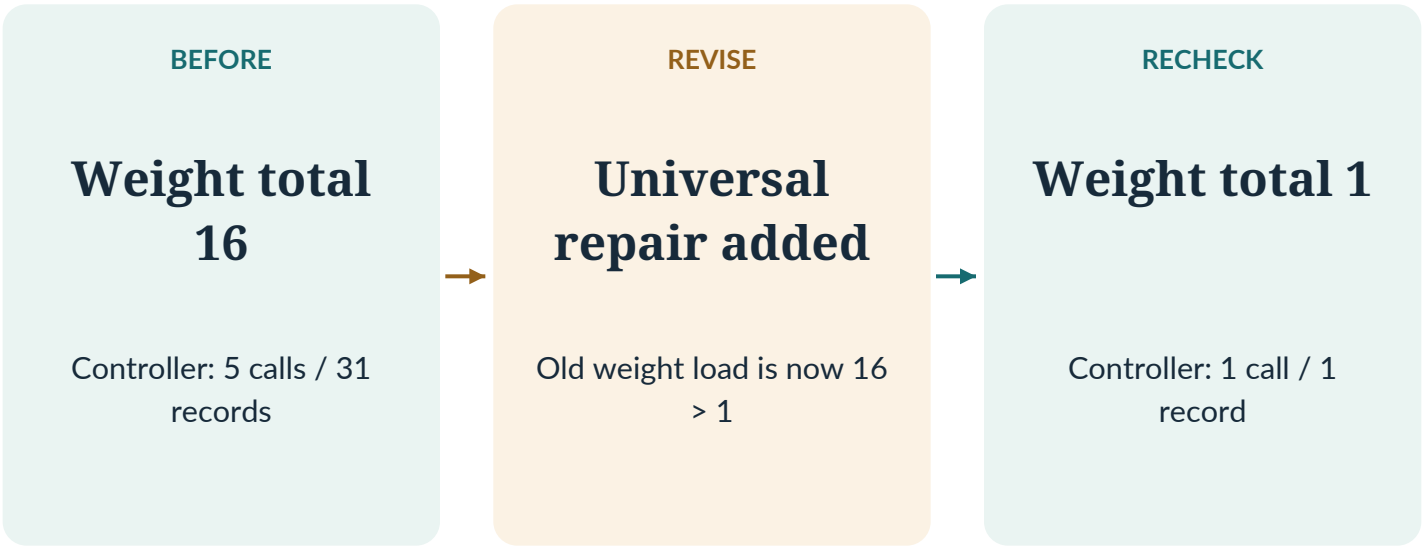
This is not automatic discovery of every dependency in arbitrary code. An omitted premise can leave a registry green while its claim becomes false. Hashes do not repair missing semantics.

[S] Theorem 24.1; §24.2 | manuscript pp. 37-38

VII / CONTINUED APPLICABILITY

A stronger repair can invalidate an old proof

The conclusion “this controller is valid” and the conclusion “this change saves work” do not automatically survive together.



Semantic result: valid

The new tree is checked under the new repair contract.

Old cost claim: blocked

It is not evidence about the new search or workload.

REPAIR AND RECHECK

For the 32-case paired model, adding a universal repair invalidates the old weight constraints. Rebuilt weights permit a one-step, one-record controller.

DO NOT INHERIT THE OLD PERFORMANCE CLAIM

The earlier assessment referred to another controller and compilation. The replacement needs a new comparison. The unaffected arithmetic equation library remains applicable.

[S] Counterexample 22.3; §24.3 | manuscript pp. 35, 38-39

VII / RESOURCE ACCOUNTING

The same controller, opposite cost conclusions

Before claiming improvement, choose a common ledger and state the prices used to combine its event counts.

Model-record reads	+1,552
Rational additions	+1,552
Capacity comparisons	−1,868

Assign prices in that same order

(1, 1, 1)	(1, 0, 1)	(0, 0, 1)
+1,236	−316	−1,868
priced units saved	priced units saved	priced units saved

THE EXACT DIFFERENCE

These are old-minus-new search events on the same 32-case weighted problem. Scope and witness checks are equal and cancel only in this isolated difference.

WHAT REMAINS TO CHARGE

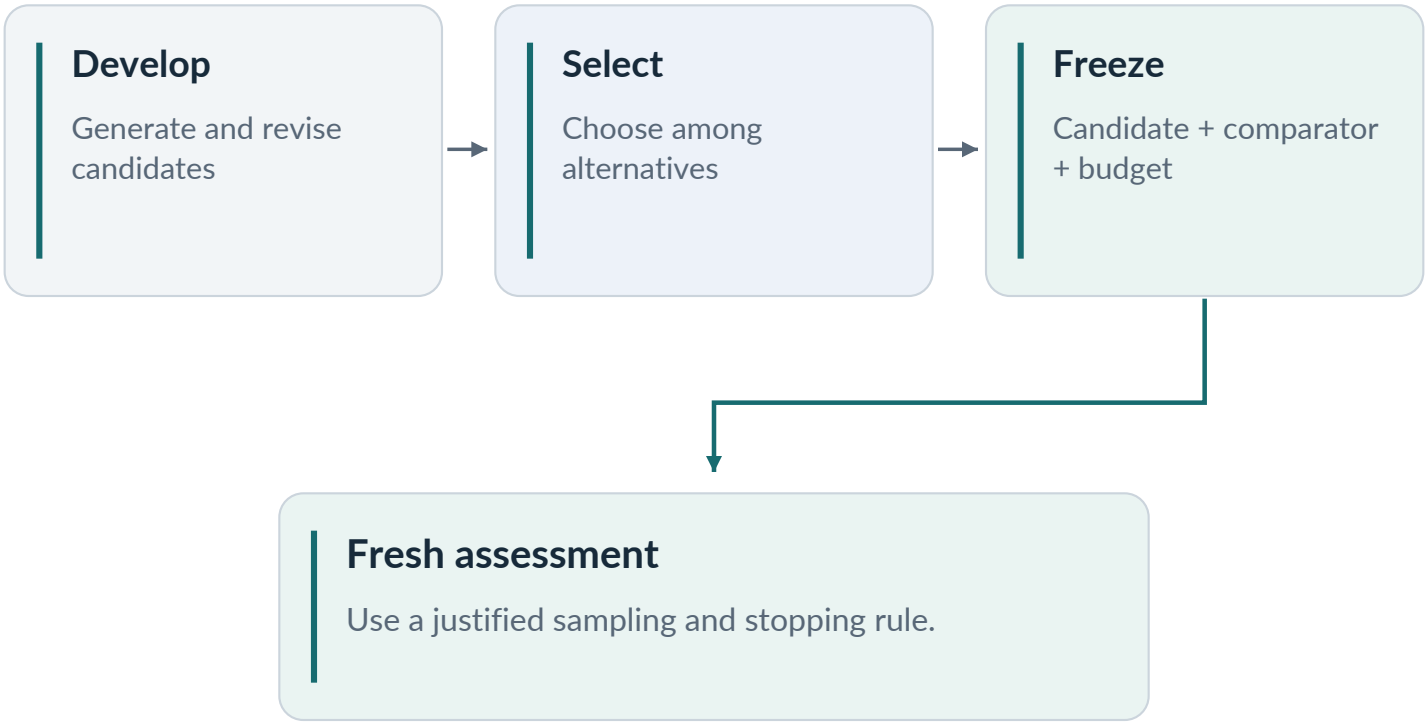
Weight preparation, equation acquisition, installed programs, storage, and revision management do not become free. Event counts alone do not determine processor time or energy.

[S] Proposition 25.1; Counterexample 25.2 | manuscript pp. 39-40

VIII / EVIDENCE AND CONTINUATION

Do not turn selection evidence into a fresh audit

What the improver has already inspected belongs to its developmental history, even when the inner solver never saw it.



Across attempts: valid conditional error allowances may be summed. Merely writing $\delta_1 + \delta_2 + \ldots \leq \delta$ does not make an invalid audit valid.

A CLAIM HAS AN EVIDENCE TYPE

A proof supports a conditional statement. A checker validates a finite witness. A fresh performance audit supports a comparison on a declared task law.

NONE SUBSTITUTES FOR THE OTHERS

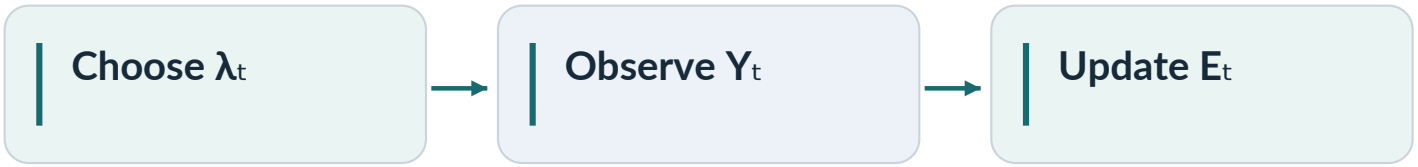
Repeatedly reading a “holdout” exposes it to the outer loop. A correct proof does not establish its physical premises. Finite algorithm counters are not independent neural trials.

[S] §26; Proposition 26.1 | manuscript pp. 41-42

VIII / EVIDENCE AND CONTINUATION

An audit can stop early without betting in hindsight

Choose the evidence update before seeing its paired score. Stopping after observing evidence is a different operation.



HINDSIGHT CONTROL: all 16 fair four-score paths

− − − −	− − − +	− − + −	− − + +
− + − −	− + − +	− + + −	− + + +
+ − − −	+ − − +	+ − + −	+ − + +
+ + − −	+ + − +	+ + + −	+ + + +

Highlighted: at least two positive scores

VALID ORDER

Freeze the comparison, choose the next bet from past information, observe the next paired score, then update the evidence process.

THE HINDSIGHT COUNTEREXAMPLE

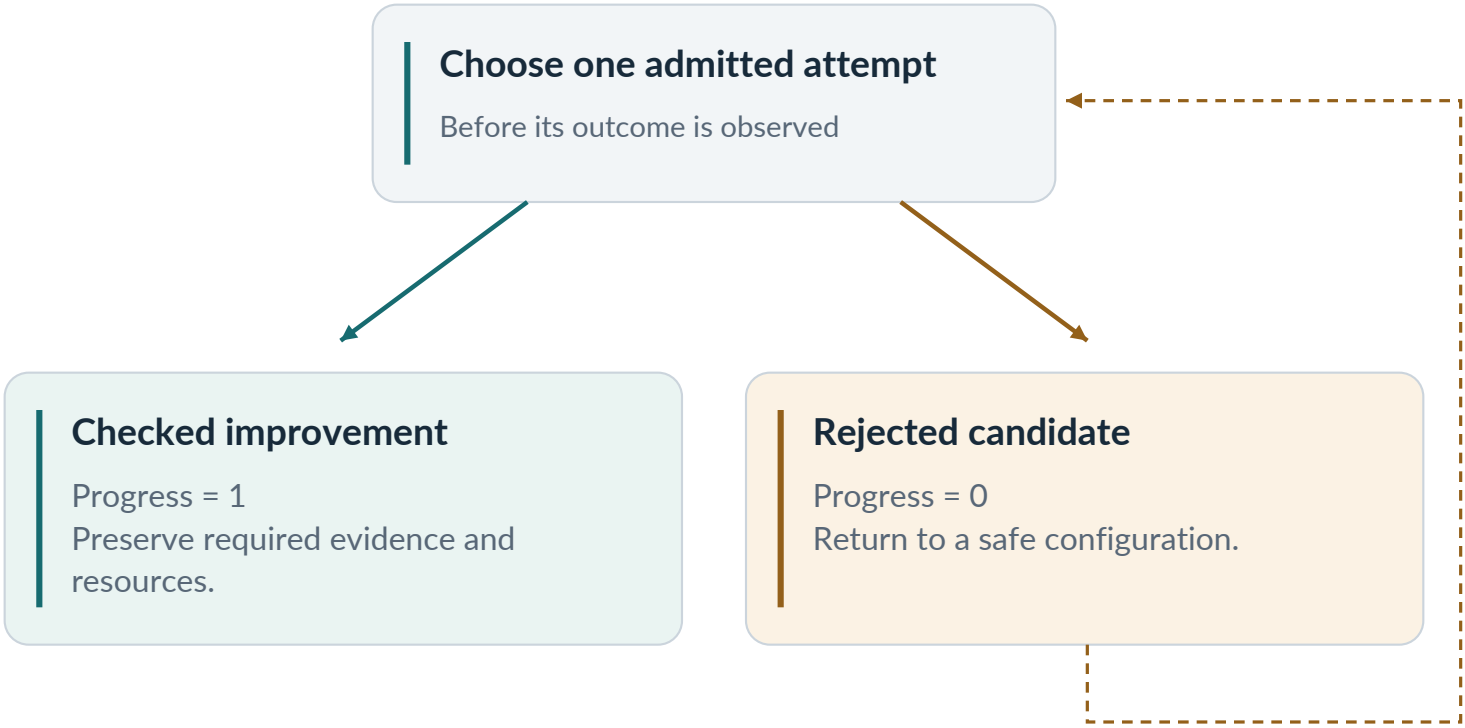
With four fair ± scores, betting only after seeing a positive doubles the evidence on positives. The nominal threshold 4 is crossed on 11 of 16 paths, not at most 1/4.

[S] Theorem 27.1; Counterexample 27.2 | manuscript pp. 42-43

VIII / EVIDENCE AND CONTINUATION

Safe repetition is not guaranteed progress

An authorized attempt must be selected before its outcome is known. One policy must meet the continuation requirement on every allowed branch.



Possible again does not mean guaranteed

This loop can repeat h times without any improvement. The supplied progress criterion cannot be replaced by “a file changed.”

THE FINITE THEOREM

Given fully observed lineage states, a finite attempt catalogue, and separately justified progress marks, the recurrence decides whether one policy guarantees k gains within h attempts.

THE REPEATED-REJECTION PATH

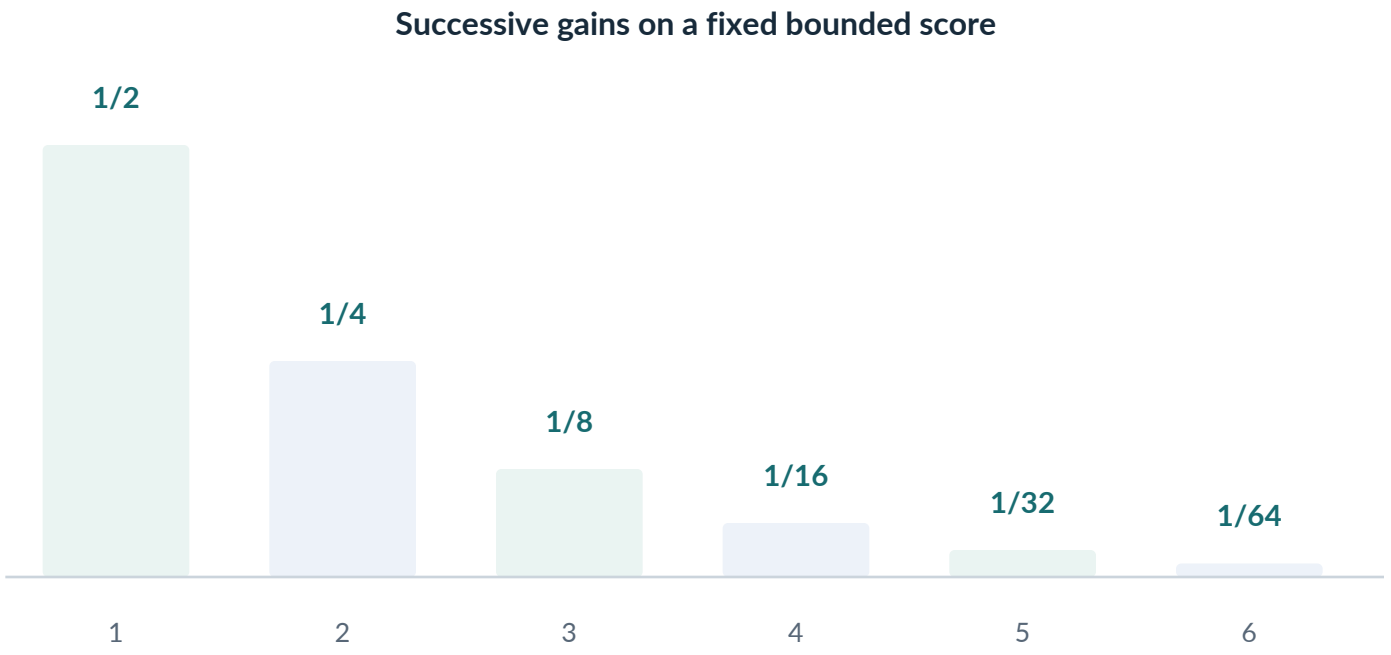
If every attempt can reject and return to the same safe state, any finite horizon admits an all-rejection path. Safety can hold without a bounded sure gain.

[S] Theorem 29.1; Counterexample 29.2 | manuscript pp. 45-46

VIII / EVIDENCE AND CONTINUATION

Positive gains need not accelerate

Recursion, continued improvement, and increasing productivity per unit resource are separate claims.



There is also a fixed-margin ceiling

On a nondecreasing score in [a,b], gains of at least $\gamma > 0$ can occur at most $\text{floor}((b-a)/\gamma)$ times. Changing the score changes the comparison.

A SIMPLE LIMITING EXAMPLE

The gains 1/2, 1/4, 1/8, … are all positive. Their accumulated total remains below 1 after every finite number of steps.

THE MISSING ADDITIONAL ARGUMENT

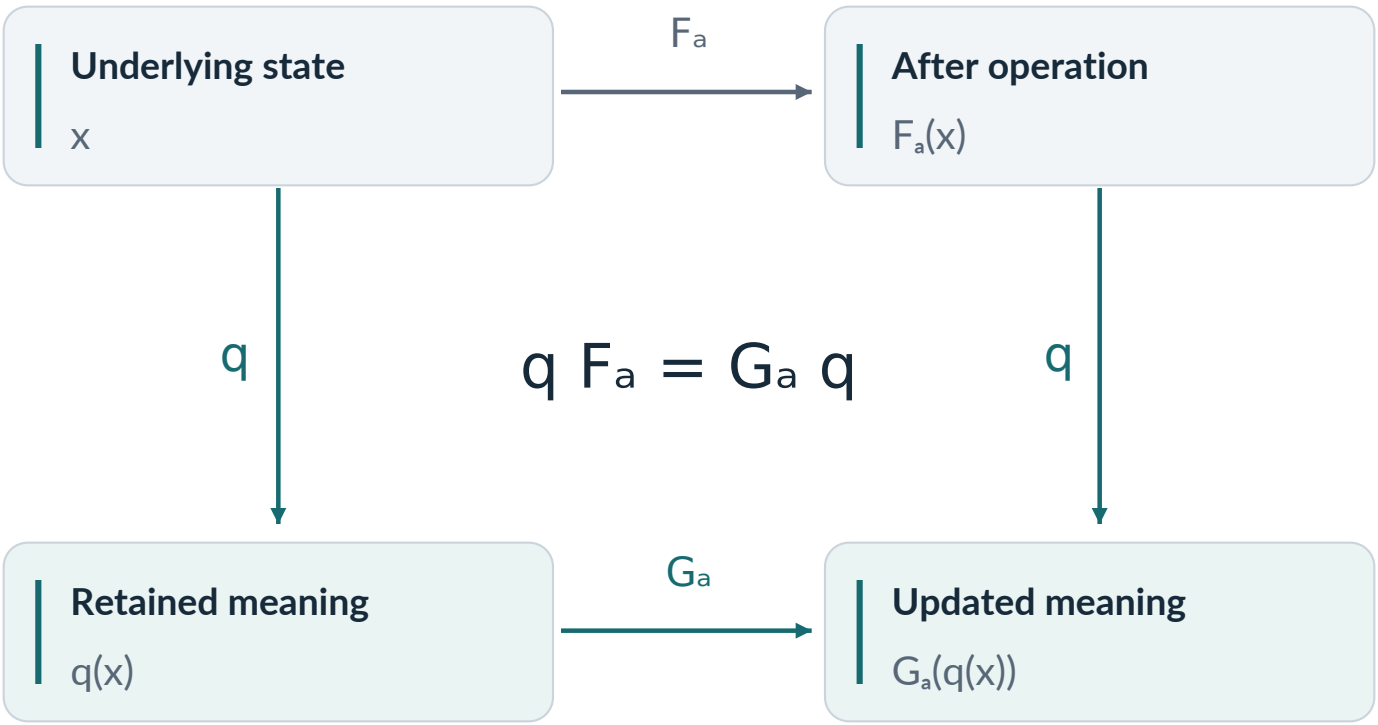
Faster improvement requires a claim about gain divided by complete discovery-and-acceptance cost. More headroom or a successful feedback loop does not establish it.

[S] Proposition 30.1; §30 | manuscript pp. 46-47

IX / WHAT REMAINS OPEN

The learned semantic bridge is still a research target

A commuting square specifies meaningful reuse. It does not discover the representation, learn the operations, or demonstrate their use by a neural system.



Specification ≠ discovery algorithm

THE CONDITIONAL RESULT

If the domain is invariant, every admitted operation commutes with q, and q preserves the required output, the relation extends to every finite operation sequence.

THE ARCHITECTURAL PROPOSAL

Learning encoders, operations, experiment selectors, and proposers is a separate problem. The complete learned Neuralese–OSL bridge is not implemented in this edition.

[S] §31; Proposition 31.1; §34 | manuscript pp. 48-49, 52-53

IX / WHAT REMAINS OPEN

Locate the recursive achievement exactly

The finite chain acquires objects at several levels. It does not erase the supplied foundations of the next outer level.

ACQUIRED OR CONSTRUCTED	STILL SUPPLIED
Diagnostic tree	Sensors and repair meanings
Improved compiler	Skeleton, bound, and checker
Pruning expression	Grammar, teacher, and features
Equations and quotient	Axioms and equation learner
Replayed certificates	Complete dependency registry

WHAT THE CONSTRUCTION SUPPORTS

A generated equation library changes the search representation used to acquire later discovery rules, while preserving the specified evidence and final programs.

WHAT IT DOES NOT CLOSE

General intelligence, learned physical semantics, autonomous invention of the primitive proof system, and indefinite improvement of the equation learner remain open.

[S] First-edition scope; §§23, 34 | manuscript pp. iii, 35-36, 52-53

REFERENCE / CONCEPTS

A small vocabulary for reading the proofs

Use these distinctions to avoid turning several different achievements into one word such as “intelligence” or “success.”

Support

The cases still possible after the reports actually received.

Repair

An action with a specified set of cases in which it succeeds.

Controller

A complete branching investigation ending in justified repairs.

Compiler

The procedure that constructs or searches for controllers.

Quotient

A representation that groups expressions using checked equivalences.

Certificate

A witness tied to a claim, its inputs, and applicable assumptions.

KEEP THE OBJECTS SEPARATE

The repair changes a target. The compiler discovers a policy. The learner can change a compiler rule. The equation library can change the learner’s candidate representation.

KEEP THE EVIDENCE SEPARATE

A witness is checked under a model. A proof has hypotheses. A resource claim has a ledger. A generalization claim needs evidence beyond semantic relabeling.

[S] Definitions 3.1, 6.1; §§5, 24-25 | manuscript pp. 3, 5-7, 37-40

REFERENCE / READING ROUTES

Return to the source at the right level

This companion prioritizes mechanisms and counterexamples. The first-edition manuscript is the authority for complete hypotheses and proofs.

GUIDE	TOPIC	MANUSCRIPT
03-07	Comparison and eight-cell diagnosis	§§3-4, 6, 32
08-12	Investigation, uncertainty, transport	§§5-7, 9
13-15	Certified compiler revision	§§11-13, 16
16-20	Synthesis and safe pruning	§§14-16, 18
21-25	Repair scope and acquired equations	§§17, 19-23
26-28	Dependencies and complete costs	§§24-25
29-32	Evidence and recursive continuation	§§26-30
33-35	Architecture, boundaries, vocabulary	§§31-35

Included in the editable guide package: the unchanged source PDF, page-to-section crosswalk, diagram-generation code, data, and review notes.

SCOPE OF THIS GUIDE

All substantive claims are restatements or worked illustrations of [S]. Source statistics are attributed to the supplied first-edition records, not presented as new learning runs.

VERIFICATION BOUNDARY

The guide's examples and numerical transcriptions were checked separately. The full original 336,867-assertion suite was not rerun to make this visual companion.

[S] Komorebi P., Self-Improving Intelligence, v1.0.0, 8 October 2026.