# WHEN CACHED PROOFS CERTIFY AN EVIDENCE BUDGET

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

A cached proof can remain valid while failing to certify a new evidence budget: repricing may make its support unaffordable even when an omitted alternative still fits. We formalize reusable budget certification in finite signed-Horn theories with canonical acquisitions charged once. Complete minimal support frontiers answer exact residual-cost queries; exact whole-proof lookup under arbitrary positive prices requires every minimal support. Under bounded prices and evidence exclusions, cache distortion equals a constrained Hamming covering radius, giving storage bounds and separate thresholds for cost approximation and status coverage. Factored representations can preserve these alternatives compactly: a family with exponentially many minimal supports has a linear-size exact solver, while tree selection can incur a linearly growing cost ratio even after duplicate purchases are removed. On a fixed ProofWriter prefix, independent exhaustive closure verifies 1,864 resource queries and 5,592 budget decisions. Retaining initial infeasibility and minimum-cardinality information resolves 4,240 decisions left unresolved by a one-support cache with its original bounds. Yet 26 feasible budgets still require an affordable witness that the cache omitted. Reusable reasoning therefore requires preserving both the conclusions that certify rejection and the alternatives that enable acceptance after resource changes.

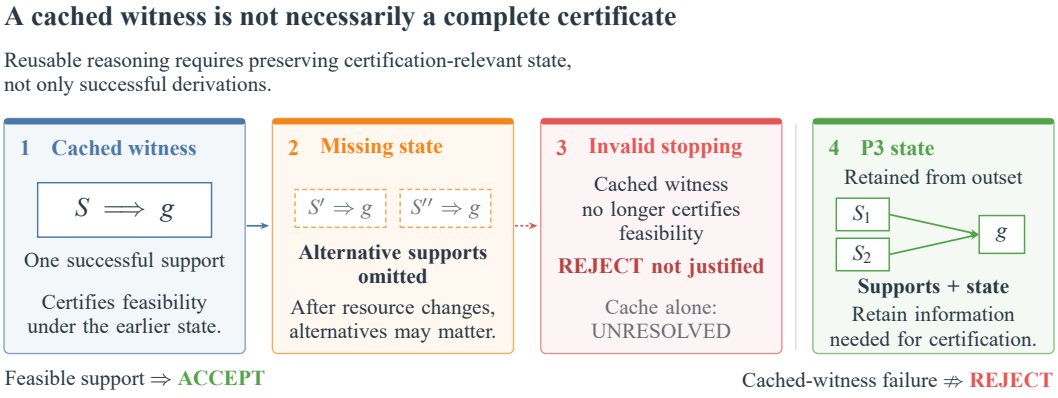


Figure 1: A cached witness certifies feasibility under an earlier resource state. When resources change, an omitted support may be needed; failure of the stored witness alone cannot certify rejection. Retaining alternative supports and resource state enables later budget certification.

## 1 INTRODUCTION

A cached proof can remain valid while giving the wrong answer to a new budget query. Suppose a system stores support $S_1 = \{a, b, c\}$, whose three acquisitions initially cost one each. The shared item $a$ is purchased once and used in two proof branches. If its price rises to five, replaying the same proof now costs seven. At budget $B = 3.5$, the cache fails. Yet an omitted support $S_2 = \{d\}$ still costs 3.5 and establishes the same goal (Figure 2). The system has lost an affordable witness, not established that the goal is unaffordable.



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

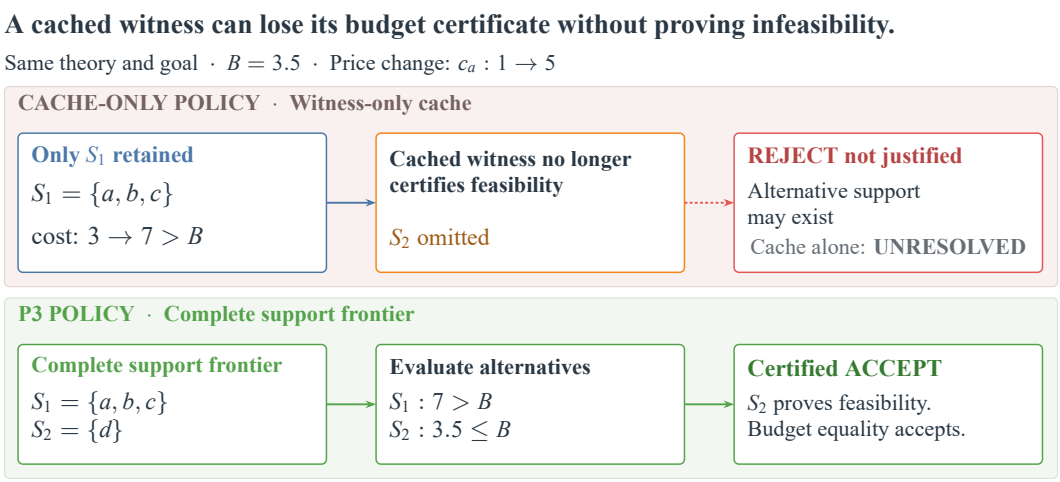


Figure 2: Repricing can invalidate a cached budget decision without invalidating the proof. With $B = 3.5$, raising $c_a$ from one to five makes $S_1 = \{a, b, c\}$ cost seven; the omitted support $S_2 = \{d\}$ still costs 3.5. A complete support frontier retains the affordable alternative. The shared acquisition $a$ is charged once despite its two uses.

This problem arises whenever proofs are reused across changing evidence prices, availability, or prior acquisitions. A proof records one sufficient set of purchases; a budget decision asks whether *any* eligible set is affordable. Acceptance needs one such witness. Rejection needs information that excludes all affordable witnesses, which may be a valid lower bound, a previously established impossibility, or an exact representation of the alternatives. The central question is therefore: *what must a reasoning cache retain to certify future resource decisions?*

We study this question in finite signed-Horn theories with canonical acquisitions charged once. The logical theory stays fixed while prices and resource states change. This separates what a stored proof establishes from what the retained representation can decide. In particular, initialization can establish more than a successful route: an exact unit-price initialization also certifies impossibility when no route exists, or a minimum number of required acquisitions when one does. Keeping those conclusions can certify later rejections without keeping additional proofs. Their value must be separated from the value of retaining alternative witnesses.

Our results characterize this distinction at three levels. *Exact reuse:* complete minimal support frontiers answer all residual-cost queries, and a fixed library queried only for affordable whole proofs is exact under arbitrary positive prices if and only if it retains every minimal support. *Limited storage:* under bounded prices and evidence exclusions, cache distortion equals a constrained Hamming covering radius. This yields storage bounds and distinct thresholds for price approximation and status coverage. *Shared structure:* factored representations can preserve the same alternatives compactly; a family with exponentially many minimal supports admits a linear-size exact solver. In contrast, tree selection can incur a linearly growing cost ratio even after duplicate purchases are removed.

A fixed ProofWriter prefix (Tafjord et al., 2021) makes the empirical distinction directly checkable. Across 1,864 resource queries and 5,592 budget decisions, retaining initial impossibility and minimum-cardinality information resolves 4,240 decisions left unresolved by a one-support cache with its original bounds. Nevertheless, 26 feasible budgets still require an affordable witness that this cache omitted. Exhaustive fact-subset closure verifies every optimum and decision. The resulting lesson is a design criterion for reusable reasoning: retain the conclusions needed to reject and the alternatives needed to accept, or preserve their structure in an exact factored representation.

## 2 Evidence-Budget Certification from Shared Proofs

A reusable proof representation must support the decisions that a system is expected to make. A budget query has an inherent asymmetry: acceptance requires one feasible acquisition support, while

rejection requires ruling out feasible alternatives. We formalize this distinction for finite theories and use it to compare complete, compressed, and factored representations.

### 2.1 Decision contract and remaining cost

Let $U$ contain $m$ canonical acquisition identities, with fact map $\ell : U \to \mathcal{A}$ into a finite set of signed ground atoms. Different identities may supply the same atom; aliases of one physical acquisition must be resolved before defining $U$. Each rule $r \in R$ has a conjunctive body $B_r \subseteq \mathcal{A}$ and one head $h_r \in \mathcal{A}$; rules are free. The requested signed goal $g$ is fixed. Signs are part of atom identities: explicit negative support is required, and neither default negation nor explosion from opposite signs is implicit. Empty rule bodies are true.

For $P \subseteq U$, define the accumulated forward closure by

$$X_0(P) = \{\ell(i) : i \in P\}, \qquad X_{t+1}(P) = X_t(P) \cup \{h_r : B_r \subseteq X_t(P)\}. \tag{1}$$

Write $\mathrm{Cl}(P)$ for its least fixed point. This permits positive cycles without permitting unsupported circular proofs.

Let $I, F, A \subseteq U$ be pairwise disjoint acquired, forbidden, and currently available identities. They need not cover $U$: an unclassified identity is not eligible for purchase. For nonnegative rational prices $c_i$, the target is

$$C_g(I, F, A; c) := \min_{\substack{Q \subseteq A \\ g \in \mathrm{Cl}(I \cup Q)}} \sum_{i \in Q} c_i, \qquad \min \varnothing := +\infty. \tag{2}$$

Set-valued purchases charge a reused acquisition once. Historical payment $s_{\mathrm{paid}}$ is separate: a finite budget $B$ accepts exactly when $s_{\mathrm{paid}} + C_g \le B$. Repricing an acquired item does not rewrite that payment.

Proof completion is neither a sum of independent branch costs nor, in general, a submodular acquisition objective. Conjunctive prerequisites create complementarity, while shared acquisitions couple otherwise distinct branches. The objective already contains minimum set cover: create one costed acquisition per subset, OR its incident acquisitions at each universe element, and AND the element gates at the goal. Thus exact repeated queries require an explicit representation or exploitable structure rather than a generic local-gain rule.

### 2.2 Foundation: complete support frontiers

Call $S \subseteq U$ a support for atom $a$ when $a \in \mathrm{Cl}(S)$, and minimal when no proper subset is also a support. For a set family $\mathcal{C} \subseteq 2^U$, let $\mathrm{Min}(\mathcal{C})$ retain its inclusion-minimal members and let $\uparrow \mathcal{C} = \{P \subseteq U : \exists S \in \mathcal{C},\ S \subseteq P\}$. The empty family denotes false; $\{\varnothing\}$ denotes a free proof. Initialize $\mathcal{M}_0(a) = \{\{i\} : \ell(i) = a\}$ and update synchronously:

$$\mathcal{C}_{r,t} = \mathrm{Min}\left\{ \bigcup_{a \in B_r} S_a : S_a \in \mathcal{M}_t(a) \text{ for every } a \in B_r \right\},$$
$$\mathcal{M}_{t+1}(a) = \mathrm{Min}\left( \mathcal{M}_t(a) \cup \bigcup_{r : h_r = a} \mathcal{C}_{r,t} \right). \tag{3}$$

An empty-body product is $\{\varnothing\}$; a nonempty body containing an empty support family contributes no support. Intermediate unions can be minimized after each body factor.

**Theorem 1** (Complete support compilation preserves every minimal alternative)**.** *Let $L = |\{h_r : r \in R\}|$. For every atom $a$, subset $P \subseteq U$, and round $t \ge 0$,*

$$P \in \uparrow \mathcal{M}_t(a) \quad \Longleftrightarrow \quad a \in X_t(P).$$

*Moreover $X_L(P) = \mathrm{Cl}(P)$ for every $P$, and $\mathcal{M}_L(a) = \mathcal{M}_{L+1}(a)$ consists of all inclusion-minimal supporting acquisition sets for $a$.*

These are the standard minimal-environment semantics of ATMS labels (de Kleer, 1986), also expressible through Boolean recursive provenance (Green et al., 2007). Here they provide the exact

support object needed by the budget interface. Every strict closure round introduces at least one previously absent rule head, which gives the uniform $L$-round bound for the grounded theory. Write $\mathcal{M}_g := \mathcal{M}_L(g)$. Appendix B.1 gives the induction and shows why unsupported cycles create no proof.

**Proposition 2** (Residual-cost query)**.** *Under the contract above,*

$$C_g(I, F, A; c) = \min_{\substack{S \in \mathcal{M}_g \\ S \cap F = \varnothing,\ S \setminus I \subseteq A}} \sum_{i \in S \setminus I} c_i. \tag{4}$$

*A minimizing support yields purchases $Q = S \setminus I$ and the final selected set $I \cup S$. The availability test can be omitted only if $I, F, A$ partition $U$.*

Nonnegative prices make subset dominance safe even after acquisitions: an unnecessary acquired item can remain in $I \cup S$ without another payment. The same support frontier therefore answers new prices and statuses without recomputing logical closure. Query normalization and a frontier scan take $O((|\mathcal{M}_g| + 1)(m + 1))$ elementary and exact rational operations. The implementation returns the exact additional cost, minimizing support, new purchases, and final set; budget equality accepts and historical payments stay separate.

Figure 2 also illustrates state reuse. Under the initial prices, acquiring $a$ for one leaves $\{b, c\}$ as the two-unit remainder of support $\{a, b, c\}$, so the original total boundary remains three. If $b$ then becomes forbidden, that support is ineligible and $\{d\}$ costs $7/2$ additionally, giving total boundary $9/2$. The compiled support frontier is unchanged; only the resource state and payment ledger change.

**Initialization conclusions are reusable certificates.** Exact unit-price initialization also supplies inexpensive information beyond a witness. If $g \notin \mathrm{Cl}(U)$, it remains impossible under every later state within the same $U$ and fixed theory. Otherwise, a certified unit-price optimum has minimum support cardinality $k_0$. Any successful completion after acquiring $I$ needs at least $\max\{k_0 - |I|, 0\}$ new acquisitions. Their cheapest possible current cost is a valid lower bound, even when no full frontier is retained. Appendix F.1 proves these statements and uses them as stronger cache controls. Such bounds can certify rejection but cannot supply an omitted acceptance witness.

### 2.3 Compression creates a certification tradeoff

The complete frontier may be expensive to retain, so consider a library that stores only selected whole-proof supports and answers a query by looking for an affordable stored entry.

**Proposition 3** (Exactness criterion for a fixed whole-proof library)**.** *In the all-available setting, let $\mathcal{L}$ contain only feasible supports. Suppose its query procedure accepts $(c, B)$ exactly when a stored $S \in \mathcal{L}$ satisfies $\sum_{i \in S} c_i \le B$. This procedure is exact for all strictly positive rational prices and nonnegative rational budgets if and only if $\mathcal{M}_g \subseteq \mathcal{L}$.*

A missing minimal support $T$ is exposed by prices 1 on $T$ and $m + 1$ outside it: every other feasible set contains an identity outside $T$, so costs at least $m + 1 > |T|$. Budget $|T|$ is therefore falsely rejected. Conversely, every feasible set contains a minimal support. The empty-support case uses budget zero; an unprovable goal has an empty library.

The proposition separates the two directions of a budget decision. Any stored support is an acceptance certificate when it is eligible and affordable. A fixed whole-proof library becomes a universal rejection basis under arbitrary positive prices only when it contains every minimal support. The same complete frontier also suffices for all status queries in Equation (4). This requirement is specific to explicit whole-proof lookup; compact implicit representations can preserve alternatives without enumerating them.

### 2.4 Approximate certification through geometric coverage

For a positive rational box $\underline{c}_i \le c_i \le \bar{c}_i$, let $\varnothing \neq \mathcal{L} \subseteq \mathcal{M}_g$ be a fixed library of nonempty supports. With all acquisitions available and none acquired, its stretch is $\Gamma(\mathcal{L}) = \max_c A_{\mathcal{L}}(c)/C_g(c)$, where $A_{\mathcal{L}}(c) = \min_{T \in \mathcal{L}} c(T)$ and $c(S) = \sum_{i \in S} c_i$. This is one plus classical relative set regret. Its

established corner formula (Crema, 2018, Lemma 6) is

$$\Gamma(\mathcal{L}) = \max_{S \in \mathcal{M}_g} \frac{\min_{T \in \mathcal{L}} c^S(T)}{\underline{c}(S)}, \qquad c_i^S = \begin{cases} \underline{c}_i & i \in S, \\ \overline{c}_i & i \notin S. \end{cases} \tag{5}$$

We use this relation to couple price distortion with changing evidence status.

Consider $g_k = \bigwedge_{i=1}^{k} (a_i \vee b_i)$ on $2k$ distinct acquisitions, with prices independently in $[1, \kappa]$, rational $\kappa > 1$, and none acquired. A minimal support is a binary word choosing one item from each pair. A status query forbids one alternative in each of at most $s$ pairs, $0 \le s \le k$, and never forbids both. The goal remains feasible. A fixed nonempty library $\mathcal{C} \subseteq \{0,1\}^k$ filters ineligible supports and selects its cheapest survivor; it cannot assemble a new support from fragments. Let $\Gamma_s$ be its worst cost ratio over these prices and statuses, with value infinity when a feasible status has no surviving entry.

A forbidden alternative fixes the opposite choice in that pair. A status therefore fixes a set $J$ of coordinates of the target word $x$; an eligible cached word must match $x$ on those coordinates. Outside $J$, a mismatch can cost $\kappa - 1$ at the price corner favoring $x$. The following radius measures exactly the worst number of paid mismatches after eligibility filtering.

**Theorem 4** (Cached-proof distortion equals constrained covering radius)**.** *Define the radius with $s$ exact coordinates by*

$$R_s(\mathcal{C}) = \max_{\substack{x \in \{0,1\}^k \\ J \subseteq [k],\, |J| = s}} \min_{\substack{y \in \mathcal{C} \\ y_J = x_J}} d_H(x, y), \qquad \min \varnothing = +\infty. \tag{6}$$

*Then $\Gamma_s(\mathcal{C}) = 1 + (\kappa - 1) R_s(\mathcal{C})/k$ exactly. Finite stretch holds if and only if $\mathcal{C}$ is a binary covering array of strength $s$, meaning that every assignment on every $s$ coordinates occurs in a cached support. For integer $0 \le r \le k - s$, let $N(k, s, r)$ be the minimum cache size achieving factor $1 + (\kappa - 1) r/k$, and write $V(n, r) = \sum_{j=0}^{r} \binom{n}{j}$. Then*

$$\left\lceil \frac{2^k}{V(k-s, r)} \right\rceil \le N(k, s, r) \le \min \left\{ 2^k, \left\lceil \frac{2k\, 2^k}{V(k-s, r)} \right\rceil \right\}. \tag{7}$$

*For $s = \lfloor \alpha k \rfloor$, $r = \lfloor \rho k \rfloor$, $0 \le \alpha < 1$ and $0 < \rho < (1 - \alpha)/2$, with binary entropy $H_2$,*

$$\lim_{k \to \infty} \frac{\log_2 N(k, s, r)}{k} = 1 - (1 - \alpha) H_2\left( \frac{\rho}{1 - \alpha} \right). \tag{8}$$

For a fixed status face, Equation (5) applies to its feasible supports and surviving library. At the corner favoring $x$, the optimum costs $k$ and a surviving $y$ costs $k + (\kappa - 1) d_H(x, y)$. Thus price robustness and availability robustness are two geometric requirements on the same codebook: cached words must first match the coordinates forced by status and then lie close to the cheapest surviving support on the remaining coordinates.

For the size bound, fix $J$ and count radius-$r$ balls within each of its $2^s$ faces. Conversely, uniformly sampled words cover each target/mask pair with probability $V(k - s, r)/2^k$; a union bound over $2^k \binom{k}{s}$ pairs proves existence at the upper bound. The counting tools are classical covering theory (Potukuchi & Zhang, 2019; Quistorff & Schlage-Puchta, 2011); the theorem gives their exact price/status interpretation for a retained whole-proof library. Appendix D proves the endpoints and rate.

**Corollary 5** (Two storage thresholds)**.** *For $s = 0$, polynomial-size whole-proof libraries have asymptotic stretch at least $(\kappa + 1)/2$, and two antipodal supports attain the limit. Requiring finite stretch for every status with at most $s$ exclusions additionally forces $|\mathcal{C}| \ge 2^s$. Hence polynomial-size libraries cover every allowed status precisely when $s = O(\log k)$; in that regime the same limiting stretch is attainable with at most $4k2^s$ stored supports. A linear fraction of exclusions forces exponential storage even before demanding a nontrivial price approximation.*

The two thresholds describe different losses: covering radius controls how expensive the best surviving cached support can be, while projected coverage controls whether any cached support survives. For a finite library of $M$ supports with finite stretch, put $n = k - s$. A binomial-tail bound sharpens the necessary radius to

$$R_s \ge \max \left\{ 0, \frac{n}{2} - \sqrt{\frac{n}{2} \ln(M/2^s)} \right\}.$$

Here $M \geq 2^s$ is already necessary; the $n = 0$ case has radius zero. This finite bound quantifies the price of accuracy above the storage needed for status coverage.

**Certified three-way decisions.** A finite approximation factor is useful only together with its declared price/status domain. For $k = 3$, $\kappa = 3$, the two antipodal supports have factors $5/3$ with no exclusion and $7/3$ with at most one exclusion; two exclusions can invalidate both while another support remains feasible. In general, a verified $G \geq \Gamma_s$ and the cheapest surviving cached cost $A_{\mathcal{C}}$ imply $A_{\mathcal{C}}/G \leq C_g \leq A_{\mathcal{C}}$. The interface therefore accepts when $A_{\mathcal{C}} \leq B$, rejects when $A_{\mathcal{C}} > GB$, and otherwise returns unresolved. At the attaining one-exclusion corner above, $C_g = 3$ and $A_{\mathcal{C}} = 7$: budget three is unresolved, while budget two is rejectable. Table 2 collects the representation-specific cases. An implementation guard is likewise an unresolved computation state rather than logical or budget infeasibility.

### 2.5 Exact reuse through structural factorization

The storage lower bounds above concern fixed libraries whose entries are complete supports. Factored representations can retain the shared structure without enumerating those supports. For the complementary-pair goal, for example, an exact circuit simply chooses the cheapest eligible item in each pair in $O(k)$ operations, with $O(1)$ cost update per pair.

More generally, shared-boundary conditioning gives an exact implicit alternative (Darwiche & Marquis, 2002). Cut the outgoing edges of every shared node, including internal gates, and condition on their Boolean values. Each remaining component is a tree; a purchase is charged only at its owning leaf. Consistent component minima join into a global purchase assignment. Theorem 7 in Appendix A gives preprocessing $O(2^q(N+E))$ and one-leaf update $O(2^q(H+1))$, with normalization assumptions and witness costs. The parameter $q$ is the shared-node count of the supplied/pruned DAG representation; cyclic unrolling can increase it sharply even when the original graph is a chain.

Direct support compilation and conditioned factoring therefore expose different structural parameters. The former serves the small acquisition universes in our ProofWriter validation and avoids unrolling-induced growth in $q$; the latter can be dramatically more succinct when a DAG has limited sharing. Preprocessing, repeated queries, updates, and witness extraction need not favor the same representation.

### 2.6 One shared acquisition separates representation and selection

A single construction makes the distinction explicit: exponentially many whole-proof supports can coexist with a compact exact factorization, while a tree objective can choose the wrong support even after its returned tree is deduplicated.

**Corollary 6** (One shared acquisition)**.** *For $k \geq 2$, let all $2k + 1$ acquisition identities be distinct and set*

$$g = \bigwedge_{i=1}^{k} \big[b_i \vee (a \wedge c_i)\big]. \tag{9}$$

*This circuit has $q = 1$ and exactly $2^k$ minimal supports. A universally exact explicit library needs all of them, while shared-boundary conditioning uses $O(k)$ preprocessing and stored scalars. After root balancing, a private-leaf update costs $O(\log k)$ and an update of $a$ costs $O(1)$; witness extraction costs $O(k)$. With prices $c(a) = c(b_i) = k^2$, $c(c_i) = 1$, the scalar-optimal tree strictly chooses all $b_i$. Its deduplicated purchase cost $U$ and the exact optimum $C$ satisfy*

$$U = k^3, \qquad C = k^2 + k, \qquad U/C = k^2/(k+1).$$

The supports are $\{b_1, \ldots, b_k\}$ and, for each nonempty index set $J$, $\{a\} \cup \{c_i : i \in J\} \cup \{b_i : i \notin J\}$. Conditioning on whether $a$ is selected leaves independent branch choices; thus

$$C = \min \left\{ \sum_i c(b_i),\ c(a) + \sum_i \min\{c(b_i), c(c_i)\} \right\}. \tag{10}$$

The tree instead minimizes each $c(b_i)$ against $c(a) + c(c_i)$ and strictly chooses the former under the stated prices. Its union has no duplicate purchases to remove. The error is therefore not merely

repeated charging: the tree objective selects the wrong proof altogether, and the ratio grows linearly with $k$ despite there being only one shared acquisition. The same exact formula handles statuses by replacing each acquired price by zero and each ineligible price by infinity; historical payments stay separate. Appendix B.5 proves all claims.

This separation isolates representation succinctness: favorable factoring can be compact even when the complete-support family is exponential. Appendix B.6 gives the exact cyclic unrolling and shows how the structural parameter can grow under a different representation.

## 3 VALIDATING REPRESENTATION-DEPENDENT CERTIFICATION

We test which budget decisions can be recovered by retaining initialization certificates, and which still need alternative witnesses. A fixed finite ProofWriter case permits exhaustive verification of both distinctions.

**Fixed inputs and exact reference.** We use the first 32 physical records of the official ProofWriter V2020.12.3 `OWA/depth-3/meta-test.jsonl` member (Tafjord et al., 2021). The release, prefix, two price vectors, two states, and resource guard were fixed before reading these records; a separate development prefix checked the formal grammar. All 466 requested signed goals enter the evaluation. The adapter grounds the supplied formal facts and rules over the finite entity domain; explicit negatives are separate atoms. Gold proofs, answer labels, and proof depths are excluded from compilation. A false-labelled goal is queried as written rather than replaced by its opposite.

Prices are either all one or $c_i = 1 + (i \bmod 4)/3$, in original fact insertion order. The initial state makes every fact available. The changed state acquires the first fact and forbids the last distinct fact; the original acquisition payment remains in $s_{\mathrm{paid}}$. Every goal serves all four conditions. Independent forward closure over all fact subsets verifies every minimal support family, all 1,864 optima (456 finite), and 5,592 budget decisions. Finite optima are tested at the exact total-cost boundary and $1/6$ on either side; infeasible queries use three finite budgets. All agree, with no resource refusal counted as infeasibility. Of the 466 initial frontiers, 338 are empty, 117 are singletons, and 11 contain multiple supports (Appendix E).

**An old optimum misses feasible budgets after repricing.** Freeze each goal's unit-price, all-available optimal support $S_0$. Under cyclic prices it becomes strictly suboptimal in eight available-state and five changed-state queries, covering eight distinct goals. Binary lookup that rejects whenever this stored support is unaffordable gives 26 false rejections among the 912 feasible budget decisions. The cached supports remain eligible in all these cases: the failure is an omitted cheaper proof. In the first strict case (record 2, question 3), the cached acquisition `triple4` costs 2 after repricing, whereas another support costs $5/3$. Budgets $5/3$ and $11/6$ are feasible, although the old proof fits neither.

**Separating an omitted witness from an omitted certificate.** All cache controls use the same frozen $S_0$ and its eligible residual cost as an upper bound. They accept an affordable cached witness, reject only when a valid total-cost lower bound exceeds the budget, and otherwise return UNRESOLVED. We compare three nested controls. The first uses the original bounds: a factor-two bound in the all-available state and historical payment in both states. The second also retains an initial infeasibility bit. If even all of $U$ cannot prove the fixed goal, no later subset of $U$ can do so. The third retains the certified initial minimum cardinality $k_0 = |S_0|$. Any successful completion needs at least $h = \max\{k_0 - |I|, 0\}$ new acquisitions, so the sum of the $h$ cheapest current available prices is a lower bound; fewer than $h$ available items certify infeasibility. It combines this bound with the original one. These controls use no new proof search or additional supports. The fourth policy retains the complete frontier. Appendix F proves the bounds and gives the exact decision rules.

**What the stronger controls resolve.** The initial infeasibility bit removes 4,056 unresolved decisions, reducing the count from 4,649 to 593. The minimum-cardinality bound resolves another 184, leaving 409 (Table 1). Thus 4,240 decisions are recovered without retaining another proof. Of the remaining 409, 383 are infeasible budgets for which these bounds do not certify rejection; 26 are feasible budgets with an omitted affordable witness. In particular, 355 unresolved cases concern

Table 1: Retaining initialization conclusions recovers rejection certificates, while omitted affordable witnesses still limit acceptance. All policies use the same 5,592 budgets; the three cache policies share the same frozen support. The second and third rows add information cumulatively. Every resolved answer is correct.

| Retained information | Accept | Reject | Unresolved |
|---|---|---|---|
| One support + original bounds | 886 | 57 | 4,649 |
| + initial infeasibility | 886 | 4,113 | 593 |
| + minimum-cardinality bound | 886 | 4,297 | 409 |
| Complete support frontier | 912 | 4,680 | 0 |

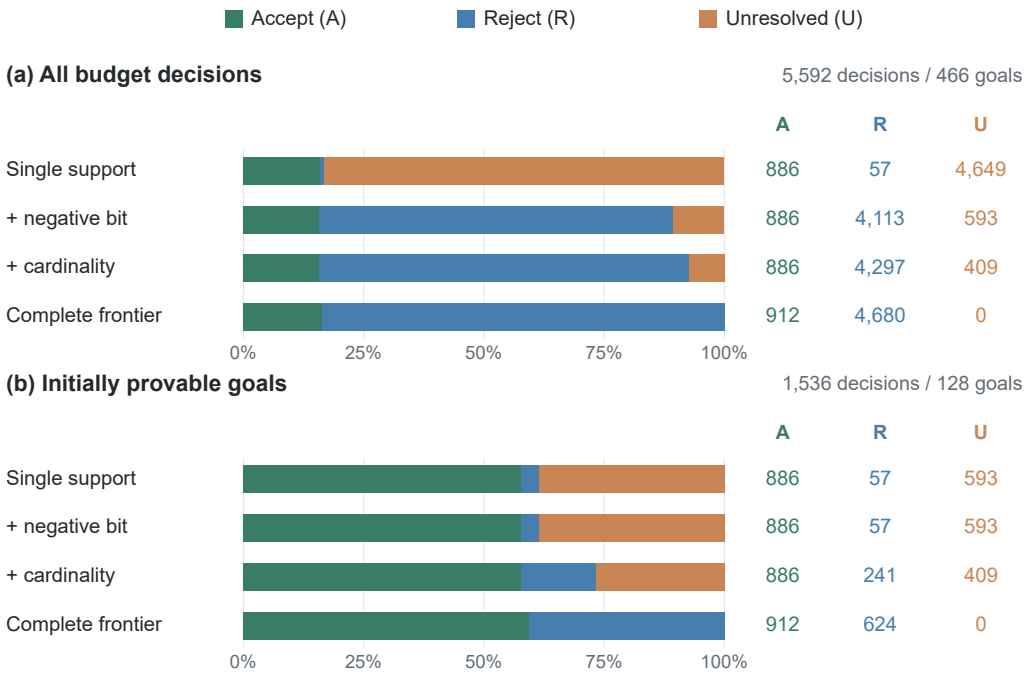


Figure 3: Certification coverage by retained initialization information. Top: all 5,592 budget decisions. Bottom: the 1,536 decisions for the 128 initially provable goals. The second and third policies add information cumulatively; all three cache policies use the same frozen support. Complete frontiers resolve every decision. Exact counts accompany the proportional bars. Every resolved answer is correct; UNRESOLVED is not a rejection.

goals whose full frontier is a singleton: storing one support without a completeness certificate does not establish that no other support exists. Coverage gains must therefore distinguish stronger rejection certificates from additional acceptance witnesses. Figure 3 shows both all budgets and the initially provable subset.

The 26 repricing failures survive both added controls. All three sound caches leave them unresolved, while complete frontiers accept them. Initialization certificates repair the large gap caused by forgotten negative information; they do not repair an omitted affordable route. The counts describe these specified policies on the fixed prefix, rather than an optimality bound for all algorithms using the same retained data.

**Accounting and proof selection are separate effects.** The scalar tree control charges leaf occurrences and keeps the first witness on equal scalar cost. Let $T$ be its occurrence cost, $U$ its deduplicated purchase cost, and $C$ the exact shared-purchase optimum. All 456 feasible queries satisfy $C \leq U \leq T$. Overcount $T > U$ occurs in 43 queries (18 goals), with maximum gap $14/3$; genuine

selection loss $U > C$ occurs in six (two goals), with maximum gap $7/3$. The other 450 tree supports are cost-optimal. These effects support the theory's separation of accounting, proof selection, and reuse. Appendix F gives the full condition ledger and the independently verified replay.

## 4 Related work

**Cost-aware reasoning with shared resources.** Charniak & Shimony (1994) study least-cost explanations with assumption and rule costs, including a MAP interpretation; here rules are free and prices apply to canonical acquisitions. Desmedt & Wang (2004) distinguish resource costs on AND/OR graphs from costs of unfolded computation. We likewise separate tree-occurrence cost from the deduplicated purchase cost.

**Reusable reasoning representations.** ATMS labels maintain minimal assumption environments by union and subsumption (de Kleer, 1986); our support frontiers use these semantics without inconsistency constraints. Recursive provenance also tracks derivation multiplicities (Green et al., 2007). Knowledge compilation and conditioning separate representation cost from query capability (Darwiche & Marquis, 2002; Mateescu & Dechter, 2005). We ask which retained alternatives still certify budget decisions after prices or evidence status change.

**Robust solution libraries and geometric coverage.** The whole-proof library stretch is one plus relative set regret, and its corner formula follows Crema (2018). Binary covering gives the sphere bound and rate (Potukuchi & Zhang, 2019); covering arrays require every projected assignment (Quistorff & Schlage-Puchta, 2011, Definition 2). Our constrained radius combines exact status matching with Hamming price distortion to obtain the budget and storage interpretation.

**Natural-language reasoning benchmarks.** ProofWriter supplies formal facts and rules alongside natural-language instances (Tafjord et al., 2021). Our allocation diagnostic uses that supplied formal structure, with fixed acquisition identities, prices, and statuses; it does not evaluate parsing, retrieval, or answer generation.

## 5 Discussion

A reusable proof cache is a representation for future decisions. An eligible affordable support certifies acceptance; a valid lower bound can certify rejection without enumerating every support. Complete frontiers or exact factored solvers provide both decisions across the full stated query domain. For a limited whole-proof library, the constrained covering radius controls price distortion, while status coverage determines whether any witness survives. These are distinct demands on what the cache preserves.

The experiment sharpens the distinction between a proof and the conclusions established while finding it. Retaining initial impossibility and minimum cardinality recovers 4,240 decisions with no additional proof supports. Yet 26 feasible budgets still lack an affordable cached witness after repricing. The two remedies serve different purposes: retain initialization conclusions to strengthen rejection, and preserve alternatives or their shared structure to support acceptance after change. An unresolved answer identifies where the retained information and decision rule supply neither certificate.

This perspective applies directly to the design of persistent reasoning state: specify the decisions that reuse must support, then retain witnesses, bounds, and structural completeness information accordingly. The finite ProofWriter case verifies these mechanisms under a sound supplied theory, canonical acquisition identities, and additive nonnegative prices. Its small frontiers make exhaustive validation possible; broader retrieval tasks may have larger frontiers, changing proof structure, or coupled purchases. Extending the representation and resource model to those settings is a substantive next step. Within the present model, the results already show why successful trace retention alone is insufficient and how compact structure can support exact reuse without an explicit list of all proofs.

## AI USE STATEMENT

Generative AI tools assisted in auditing mathematical derivations for rigor, refining manuscript language and readability, and supporting code implementation and debugging. The authors take full responsibility for the correctness of all mathematical results, experimental findings, code, and the final manuscript.

## REPRODUCIBILITY STATEMENT

Appendices B–D give full proofs and finite theorem checks. The fixed-case protocol pins the dataset release, physical prefix, acquisition identities, prices, and states. The revised artifact includes every saved goal ledger, an exact rational replay of all four cache policies, and an independent verifier that reparses the source facts and rules and enumerates all fact subsets. It verifies all minimal support families, optima, budget labels, and resolved cache decisions without calling the producer or replay implementation. The same commands regenerate the result tables and coverage figure. No trained model is required.

## ETHICS STATEMENT

This study uses public, synthetic formal reasoning records and reports no human-subject experiment or deployment. Its guarantees assume a sound supplied theory and an explicit acquisition model. A formally supported goal need not be a true statement about the world. Applying budget rejection to incomplete or incorrect evidence can suppress useful information; the stated completeness boundary must accompany any such application.

## REFERENCES

Eugene Charniak and Solomon Eyal Shimony. Cost-based abduction and MAP explanation. *Artificial Intelligence*, 66(2):345–374, 1994. doi: 10.1016/0004-3702(94)90030-2.

Alejandro Crema. Min max (relative) set-regret combinatorial optimization. Optimization Online preprint, 2018. URL https://optimization-online.org/wp-content/uploads/2018/10/6897.pdf.

Adnan Darwiche and Pierre Marquis. A knowledge compilation map. *Journal of Artificial Intelligence Research*, 17:229–264, 2002. doi: 10.1613/jair.989. URL https://arxiv.org/abs/1106.1819.

Johan de Kleer. An assumption-based TMS. *Artificial Intelligence*, 28(2):127–162, 1986. doi: 10.1016/0004-3702(86)90080-9. URL https://dekleer.org/Publications/An%20Assumption-Based%20TMS.pdf.

Yvo Desmedt and Yongge Wang. Analyzing vulnerabilities of critical infrastructures using flows and critical vertices in AND/OR graphs. *International Journal of Foundations of Computer Science*, 15(1):107–125, 2004. URL https://webpages.charlotte.edu/yonwang/papers/ijfcs03.pdf.

Todd J. Green, Grigoris Karvounarakis, and Val Tannen. Provenance semirings. In *Proceedings of PODS*, pp. 31–40, 2007. doi: 10.1145/1265530.1265535. URL https://www.cs.ucdavis.edu/~green/papers/pods07.pdf.

Robert Mateescu and Rina Dechter. AND/OR cutset conditioning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2005. URL https://ics.uci.edu/~dechter/publications/r121.pdf.

Aditya Potukuchi and Yihan Zhang. Improved efficiency for covering codes matching the sphere-covering bound. arXiv:1902.07408, 2019. URL https://arxiv.org/abs/1902.07408.

Jörn Quistorff and Jan-Christoph Schlage-Puchta. On generalized surjective codes. *Studia Scientiarum Mathematicarum Hungarica*, 48:75–92, 2011. doi: 10.1556/SScMath.2009.1140. URL https://doi.org/10.1556/SScMath.2009.1140.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. ProofWriter: Generating implications, proofs, and abductive statements over natural language. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 3621–3634, 2021. doi: 10.18653/v1/2021.findings-acl.317. URL https://aclanthology.org/2021.findings-acl.317/.

**Appendix Overview.** Appendix A introduces the conditioned DAG representation and its exact certification guarantees. Appendices B–D provide proofs for the theoretical results. Appendices E–F describe the experimental protocol, reproducibility details, and additional cache diagnostics.

# A Exact Certification via Conditioned DAG Representations

This appendix develops the conditioned-DAG representation used in Section 2.5 and explains its exact budget certification and localized updates.

For a supplied monotone AND/OR DAG, classical conditioning offers another realization of the same payment contract (Darwiche & Marquis, 2002). Normalize repeated input IDs, prune nodes without a path to the goal, and let $S$ contain *all* nodes of fanout greater than one, including internal gates. Each canonical purchase has one original owning leaf.

The following result shows when retained shared structure answers future budget queries exactly without listing every complete support.

**Theorem 7** (Exact budget certification and updates by shared-boundary conditioning)**.** *Let the resulting finite simple DAG have $N$ nodes, $E$ edges, and $q = |S|$. Balance high-fanin gates using fanout-one gates, retaining each original gate. Cut outgoing edges of $S$ and replace them by fixed Boolean boundary copies. Let $H$ be the maximum owning-leaf-to-component-root depth afterwards. Conditioning on all $2^q$ boundary assignments gives exact minimum additional costs with $O(2^q(N+E))$ preprocessing and stored scalars. Changing one leaf's status or nonnegative rational price updates the exact minimum in $O(2^q(H+1))$ elementary and rational operations. Witness extraction separately costs $O(N+E)$.*

The cut graph is a forest. Each assignment fixes its shared roots and the true goal; tree dynamic programming charges each leaf only at its owner. An update changes one component's ancestor path. Finite sums and counts of infinite component costs avoid undefined infinity subtraction. These are specializations of conditioning and incremental tree DP, not a new general compilation framework. Bit complexity also depends on current price encodings.

A global purchase assignment restricts to consistent components; conversely, component minima with matching boundary values join into a valid assignment. Shared internal gates must therefore be conditioned too. Appendix B gives the recurrences and consistency proof.

Complete support frontiers can have $\binom{m}{\lfloor m/2 \rfloor}$ members; even a small acyclic theory can require exponentially many explicit supports. Finite Horn unrolling is truth- and cost-exact in $L$ rounds, but may greatly increase $q$: a chain with original $q = 0$ can have $h(h-1)/2$ shared nodes after $h$-round unrolling and reachability pruning alone. Neither exact representation implies scalable end-to-end compilation. Resource refusal is not budget infeasibility. Appendix B gives full proofs, construction bounds, and edge cases.

## A.1 Implementation and decision summary

**Compile the complete support function.** Each identity occupies one bit. Rule bodies combine previous-round frontiers by bitwise union and subset absorption, implementing Equation (3). Different identities supplying one literal remain alternatives. Empty bodies contribute the empty support; unsupported atoms have an empty family. Synchronous propagation stops at a fixed point or the distinct-head bound $L$.

Table 2: Certificates and rejection guarantees under a sound supplied theory. A candidate cost above the remaining budget does not itself justify rejection; factor guarantees apply only in their declared price/status domain. Complete representations support exact decisions, subject to their construction cost.

| Representation | Available certificate | Rejection guarantee |
|---|---|---|
| Incomplete library without a lower bound | Upper bound from eligible stored purchases | Unresolved; an omitted support may be affordable |
| Library with verified factor $G$ | Interval $[A_{\mathcal{L}}/G, A_{\mathcal{L}}]$ in its declared price/status domain | Reject only if $A_{\mathcal{L}} > GB$; otherwise unresolved |
| Scalar tree with its purchase identities | Union cost $U$; occurrence sum $T \geq U$ is also an upper bound | Unresolved; a scalar or union gap can conceal an affordable support |
| All minimal supports | Exact optimum after state filtering | Exact rejection by Proposition 2 |
| Complete conditioned DAG | Exact optimum over consistent boundary assignments | Exact rejection by Theorem 7 |

# B Proofs of Support Frontier and Factored Certification Results

This appendix supplies proofs for the support-frontier and factored-certification results of Section 2.

All results use finite canonical acquisition identities, finite ground rules, and the forward interpretation in Section 2.1. Explicit opposite signs are distinct supported atoms. There is no default negation, contraposition, or implicit explosion. Prices are nonnegative exact rationals; rules have no acquisition cost. The following proofs establish the local contract using classical provenance and conditioning principles (Green et al., 2007; Darwiche & Marquis, 2002). They do not establish a new general semantics or a theorem-level identification with other cost-based explanation models (Charniak & Shimony, 1994).

## B.1 Uniform closure depth and complete support antichains

**Lemma 8** (Uniform closure bound for every future acquisition set)**.** *For $H_R = \{h_r : r \in R\}$ and every $P \subseteq U$, $X_L(P) = \mathrm{Cl}(P)$ with $L = |H_R|$. For one fixed $P$, $|H_R \setminus X_0(P)|$ rounds suffice.*

*Proof.* The sets $X_t(P)$ increase. Every newly added atom is in $H_R \setminus X_0(P)$. If $X_{t+1}(P) = X_t(P)$, applying the same closure operator again changes nothing, so all later rounds are equal. Therefore every round before stabilization must add a distinct previously absent head. There are at most $|H_R \setminus X_0(P)| \leq L$ such additions. The closure is already represented at that round; an additional procedural equality check is unnecessary for the bound.

This includes empty bodies. With no acquired seeds, the chain $\varnothing \to a_1, a_1 \to a_2, \ldots, a_{L-1} \to a_L$ first derives $a_L$ at round $L$, attaining the bound. Direct supplied facts are available at round zero. If $R = \varnothing$, then $L = 0$. A smaller bound based on a particular acquired set remains valid only while those acquisitions are required; the uniform $L$ is valid for every later status assignment on the fixed theory. □

*Proof of Theorem 1.* For finite set families $\mathcal{C}, \mathcal{D} \subseteq 2^U$, we have

$$\begin{aligned} \uparrow \mathrm{Min}(\mathcal{C}) &= \uparrow \mathcal{C}, \\ \uparrow (\mathcal{C} \cup \mathcal{D}) &= (\uparrow \mathcal{C}) \cup (\uparrow \mathcal{D}), \\ \uparrow \{S \cup T : S \in \mathcal{C}, T \in \mathcal{D}\} &= (\uparrow \mathcal{C}) \cap (\uparrow \mathcal{D}). \end{aligned} \tag{11}$$

For the first identity, every member of a finite family contains a minimal member. The second follows from existential membership. The third follows from $S \cup T \subseteq P$ if and only if both $S \subseteq P$ and $T \subseteq P$. Iterating the third identity gives the body conjunction; the empty-body product $\{\varnothing\}$ has upward closure $2^U$.

At round zero, $P \in\uparrow \mathcal{M}_0(a)$ exactly when some purchased $i \in P$ supplies $a$, which is $a \in X_0(P)$. Suppose the claim holds at round $t$. Equation (11) implies that $P \in\uparrow \mathcal{C}_{r,t}$ exactly when every body atom lies in $X_t(P)$. Taking the union over applicable heads and the previous family gives Equation (1). This proves the assertion simultaneously for every atom and purchase subset, by induction.

Every finite upward-closed family has a unique antichain representation. Indeed, if antichains $\mathcal{C}$ and $\mathcal{D}$ have the same upward closure, then for $S \in \mathcal{C}$ there exist $T \in \mathcal{D}$ and $S' \in \mathcal{C}$ with $S' \subseteq T \subseteq S$. The antichain property forces $S' = S$ and hence $T = S$. The reverse inclusion follows symmetrically. Lemma 8 makes the represented truth function constant after round $L$ for all $P$. Uniqueness proves that $\mathcal{M}_L(a)$ is its full family of minimal true purchase sets and that $\mathcal{M}_{L+1}(a) = \mathcal{M}_L(a)$.

Pruning during a body fold is also sound: if $S' \subseteq S$, then $S' \cup T \subseteq S \cup T$ for every subsequent support $T$. Consequently a discarded dominated support cannot be necessary to represent any minimal union. This is monotonicity of upward closures, not monotonicity of antichain membership: a new smaller support can replace an older member. □

For example, $a \to b$ and $b \to a$ without supplied support initialize both families to $\varnothing$, and neither rule can create one. A supplied $a$ provides a singleton support that propagates to $b$. A free rule $\varnothing \to a$ instead provides the empty support, which dominates every nonempty alternative. A proof of the opposite signed atom is never substituted for a requested goal. These facts hold without a dataset depth bound or a selected proof annotation.

### B.2 Exact residual costs and the external payment ledger

*Proof of Proposition 2.* Take an admissible $S \in \mathcal{M}_g$ in Equation (4). Then $Q = S \setminus I \subseteq A$. Because $S \subseteq I \cup S$ and forward Horn derivability is monotone, $g \in \mathrm{Cl}(I \cup Q)$. This realizes the displayed additional cost and proves that Equation (2) is no larger.

Conversely, any feasible $Q \subseteq A$ gives $g \in \mathrm{Cl}(I \cup Q)$. By Theorem 1, some $S \in \mathcal{M}_g$ satisfies $S \subseteq I \cup Q$. Since $I, F, A$ are pairwise disjoint, $S \cap F = \varnothing$ and $S \setminus I \subseteq Q \subseteq A$. Nonnegative prices imply

$$\sum_{i \in S \setminus I} c_i \le \sum_{i \in Q} c_i.$$

Thus restriction to minimal supports loses no optimum. The two constructions also show that one feasible family is empty exactly when the other is empty.

If $I, F, A$ partition $U$, avoiding $F$ implies $S \setminus I \subseteq A$. Otherwise it does not: for a sole acquisition $x$ proving $g$, take $I = F = A = \varnothing$. Filtering only against $F$ admits $\{x\}$ although no purchase is eligible. The explicit availability test is therefore necessary for the more general contract. □

An acquired fact omitted from a minimal support still remains in the selected set $I \cup S$. Dropping a dominated support cannot require a refund, remove that fact, or make it payable again. Zero prices can make larger supports tie, but a minimal support still attains an optimum. Negative prices would invalidate the subset-cost inequality and are excluded. The ledger $s_{\mathrm{paid}}$ records actual past payments; a new price vector changes only the remaining optimization. Exact equality to a finite total budget is accepted, while $+\infty$ is rejected at every finite budget.

### B.3 Separating prices and explicit-library completeness

*Proof of Proposition 3.* If $\mathcal{M}_g \subseteq \mathcal{L}$, then any feasible purchase set $P$ contains a minimal support $T \in \mathcal{M}_g$. Its cost is no greater than that of $P$, so the library accepts every feasible price–budget query. Because every library member is feasible, it cannot give a false acceptance.

For necessity, suppose $T \in \mathcal{M}_g \setminus \mathcal{L}$. Set

$$c_i = \begin{cases} 1, & i \in T, \\ m+1, & i \notin T. \end{cases} \qquad B = |T|. \tag{12}$$

The feasible set $T$ costs $|T| \leq m$. Any different feasible $P$ cannot be a subset of $T$, since that would contradict the inclusion-minimality of $T$. Hence $P$ includes some identity outside $T$ and costs at least $m + 1 > |T|$. Thus $T$ is the unique optimal feasible purchase set, and every stored member costs more than $B$. The true query accepts while the library rejects. All exposing prices are strictly positive integers, so they are included in the claimed rational-price domain.

If $T = \varnothing$, it costs zero and every different feasible set costs strictly more; $B = 0$ gives the same conclusion. For $m = 0$ there is no different subset. If the goal is wholly unprovable, both $\mathcal{M}_g$ and every library of feasible sets are empty, and the equivalence holds vacuously. □

The necessity is for the stated procedure, which tests its stored candidates. An implicit circuit or an algorithm deriving additional supports need not enumerate every minimal support in advance. Multiple proofs with the same support count as one acquisition set, whereas a complete price-independent library may require many different supports. A single stored proof can be complete when it contains the unique minimal support; incompleteness is not inferred merely from a library's name or origin.

### B.4 Shared-boundary DAG conditioning

Here the input is a finite simple monotone AND/OR DAG, with edges from an input to its consumer and one goal. Purchase leaves and logical zero-input gates are different node types. Repeated gate inputs are deduplicated by Boolean idempotence. Distinct physical aliases must already have a single canonical owner. Prune vertices without a path to the goal before measuring $N, E$ and the set $S$ of fanout-greater-than-one vertices.

**Balanced expansion preserves sharing.** Replace a $k > 2$ input AND or OR by a balanced tree of the same associative operator, retaining the original gate as its root and using each distinct input once. This adds $k-2$ gates of fanout one. Retain original zero- and one-input gates as well. Original fanouts do not change, so $S$ and $q$ are unchanged. The expanded graph has $O(N + E)$ vertices and edges. Balancing an individual gate does not make an arbitrary whole graph logarithmic in depth.

**The cut graph is a forest.** Cut every outgoing edge of $S$; replace its source at the receiving gate by a fresh fixed-value boundary leaf for that edge. Among original and expansion vertices the remaining outdegree is at most one, and the graph is acyclic. An undirected cycle of length $k$ would have $k$ directed edges, with each of its $k$ vertices contributing at most one outgoing cycle edge. Each vertex would therefore contribute exactly one, forcing a directed cycle, a contradiction. The undirected graph is a forest. Its component roots are exactly $S$ and the goal: root relevance gives every other vertex an outgoing edge, while the cuts remove all outgoing edges of $S$. Each component has one such sink. Adding separate boundary leaves preserves its tree structure.

**Two-state tree costs.** Enumerate $b \in \{0,1\}^q$. Each boundary copy of $s \in S$ is fixed to $b_s$ at zero cost. An owned purchase leaf $i$ has state costs

$$(D_i(0), D_i(1)) = \begin{cases} (\infty, 0), & i \in I, \\ (0, c_i), & i \in A, \\ (0, \infty), & i \notin I \cup A. \end{cases} \tag{13}$$

A boundary fixed to zero has costs $(0, \infty)$, and one fixed to one has costs $(\infty, 0)$. For binary child cost pairs $u = (u_0, u_1)$ and $v = (v_0, v_1)$, the exact recurrences are

$$\begin{aligned} D_\wedge(1) &= u_1 + v_1, & D_\wedge(0) &= \min\{u_0 + \min(v_0, v_1),\ u_1 + v_0\}, \\ D_\vee(0) &= u_0 + v_0, & D_\vee(1) &= \min\{u_1 + \min(v_0, v_1),\ u_0 + v_1\}. \end{aligned} \tag{14}$$

The zero-input identities are $(\infty, 0)$ for AND and $(0, \infty)$ for OR; one-input gates copy their child's pair. Arithmetic uses $\infty + x = \infty$ and never subtracts infinities. These formulas enumerate exactly the child assignments giving the requested output.

Let $\mu_C(b)$ be the minimum cost of component $C$, constrained to output $b_s$ if its root is shared node $s$, and to output one if its root is the goal. The candidate global value is

$$\min_{b \in \{0,1\}^q} \sum_C \mu_C(b). \tag{15}$$

*Exactness part of Theorem 7.* Take any globally feasible assignment of purchased leaves. Its unique circuit valuation defines $b$ at the shared vertices. Restrict its original purchases to each component. All boundary and root constraints are respected. Original purchase owners occur in exactly one component, so component costs sum to the assignment's additional cost. Replacing those local choices by their minima cannot increase it. Thus Equation (15) is no larger than the true optimum.

Conversely, take a boundary assignment with a finite sum and choose minimizing local assignments. Unite the true owned purchase leaves. By topological induction in the original expanded DAG, each vertex's global value agrees with its local value. For an uncut predecessor this follows within its tree; for a cut predecessor $s$, its owning component root and every boundary copy are all constrained to $b_s$. The goal is therefore true. Every original leaf satisfies its status constraint, and each available true leaf is paid for once at its owner. The union has exactly the summed additional cost. This proves the reverse inequality and equivalence of infeasibility.

Nonnegative prices allow irrelevant available leaves removed by pruning to remain unpurchased. Irrelevant acquired leaves remain acquired outside the goal calculation and their past payment stays in the ledger. Boundary leaves never incur or duplicate an acquisition payment. □

**Compilation, updates, and witnesses.** Cache both cost states and the constrained component contribution for every node and boundary assignment. One tree pass per assignment takes $O(N+E)$ scalar operations and stored values; the total is $O(2^q(N+E))$. Minima admit witness backpointers, or can be reconstructed by checking the constant-size child choices in Equation (14).

*Update and complexity parts of Theorem 7.* An original relevant purchase leaf belongs to one component and has a unique ancestor path to its root. A status or price change modifies only its leaf pair and the pairs on this path. All sibling subtrees and other components have the same conditional inputs. In particular, if the purchase leaf is itself shared, its owning component is a single vertex: its boundary copies remain fixed to $b_s$, and changing its status modifies only the owning component's contribution for each assignment.

For each $b$, maintain

$$f_b = \sum_{C:\mu_C(b)<\infty} \mu_C(b), \qquad z_b = |\{C : \mu_C(b) = \infty\}|.$$

To replace one contribution, subtract the old value only when it is finite; otherwise decrement $z_b$. Insert the new contribution by the corresponding finite addition or count increment. The total equals $f_b$ if $z_b = 0$ and $\infty$ otherwise. This handles multiple impossible components and $\infty$-to-$\infty$ replacement without undefined arithmetic.

There are at most $H+1$ changed node pairs per assignment, one contribution replacement, and a final scan of $2^q$ totals. The update therefore uses $O(2^q(H+1))$ elementary and exact rational operations. An irrelevant leaf's update does not alter these costs. Given a best finite assignment, traversing its component trees reconstructs all original purchases in $O(N+E)$ work, which is separate from the value-update bound. Zero costs and tied minima affect only the choice among valid witnesses. □

All finite costs sum subsets of current leaf prices. Reduced rational arithmetic prevents historical updates from creating artificial denominator growth; bit complexity still depends on numerator and denominator lengths. The bounds count scalar operations and stored values, not bit operations or bytes. They exclude topology or identity changes, hidden coupled costs, and any online competitive guarantee.

### B.5 THE ONE-SHARED-ACQUISITION SEPARATION

*Proof of Corollary 6.* Only $a$ has fanout greater than one, so $q = 1$. The circuit has $2k+1$ purchase leaves, $k$ inner AND gates, $k$ OR gates, and one root: $N = 4k+2$, $E = 5k$. Balancing the $k$-input root adds $k-2$ fanout-one gates, giving $5k$ nodes and $6k-2$ edges before cutting. The boundary copies require only $k$ additional constant-size references.

**Minimal supports.** A support without $a$ must contain every $b_i$; its only minimal possibility is $B = \{b_1, \ldots, b_k\}$. A support containing $a$ must contain at least one of $b_i, c_i$ at every index. Minimality excludes both at the same index, since either can be removed while the other satisfies that branch. Some $c_i$ must occur, since otherwise $a$ is redundant. Thus every minimal support containing $a$ is

$$S_J = \{a\} \cup \{c_i : i \in J\} \cup \{b_i : i \notin J\}, \qquad \varnothing \neq J \subseteq \{1, \ldots, k\}.$$

Conversely, deleting any pair representative from $S_J$ breaks its branch, and deleting $a$ breaks all branches indexed by $J$. Hence these sets and $B$ are minimal and distinct. There are $1+(2^k-1) = 2^k$ of them. Proposition 3 then forces every support into an explicit universally exact library. If stored as identity lists, their total length is $k + (k+1)(2^k - 1) = (k+1)2^k - 1$. This count is specific to explicit enumeration, not a bound on every representation of the family.

**Conditioned representation and updates.** Cutting $a$ leaves two components: its singleton owner and the remaining rooted tree. There are two boundary assignments. By Theorem 7, preprocessing and stored scalars are $O(k)$. An owning private leaf is at depth at most $2 + \lceil \log_2 k \rceil$ after balancing. Its price/status update therefore uses $O(\log k)$ scalar operations, including the minimum over the two totals. Updating $a$ changes only the singleton owner, costing $O(1)$ such operations. Its already fixed boundary copies do not need to be updated within an assignment. Witness extraction still costs $O(k)$. These are exact scalar-operation bounds; rational arithmetic also depends on current price encoding lengths.

**Exact costs and strict purchase-selection loss.** For any nonnegative prices, a proof avoiding $a$ requires all $b_i$ and costs at least $\sum_i c(b_i)$, which that set attains. If $a$ is bought, every branch can independently choose its cheaper representative $b_i$ or $c_i$. This gives the second term in Equation (10). It can describe a nonminimal purchase set when all $b_i$ are chosen, but nonnegative $c(a)$ prevents that case from beating the first term. Together the two alternatives prove the exact formula.

The unfolded tree objective instead is $T = \sum_i \min\{c(b_i), c(a) + c(c_i)\}$. For the stated prices, every branch strictly chooses $b_i$ because $k^2 < k^2 + 1$. Thus its selected purchase set is $B$, with no repeated acquisition; $T = U = k^3$. The exact shared choice is $\{a, c_1, \ldots, c_k\}$, with cost $k^2 + k$. It is strictly cheaper than $B$ for all $k \geq 2$, since $(k-1)k^2 > k$. Therefore $U/C = k^2/(k+1) = k-1+1/(k+1)$ grows linearly. There is no local tree tie to resolve differently, and deduplication does not alter its purchases. □

**Status-aware form.** Define $r_x = 0$ for acquired identities, $r_x = c(x)$ for available ones, and $r_x = +\infty$ for every ineligible identity. Then the remaining optimum is

$$\min \left\{ \sum_i r_{b_i},\ r_a + \sum_i \min\{r_{b_i}, r_{c_i}\} \right\}.$$

For either finite term, buy exactly its available representatives; the already acquired set remains present, so the goal is true. Conversely, any feasible selected set either contains $a$ and one representative per branch, or omits $a$ and contains all $b_i$. Its additional cost is at least the corresponding term by nonnegativity. The first candidate is a proof that uses all $b_i$; it does not require an acquired $a$ to become absent. This proves both inequalities even with acquisitions outside the chosen proof. An infinite term represents an unavailable choice, and historical payments remain external. The construction is a specialization of classical compact representation and conditioning, not a new general compilation algorithm.

### B.6 Finite unrolling and representation growth

**Lemma 9** (Finite unrolling preserves derivability under cycles)**.** *The signed Horn contract admits a finite monotone DAG with one original purchase leaf per identity that has the same goal truth for every $P \subseteq U$, and hence the same unique-payment objective under every admissible state and price vector.*

*Proof.* Create atom nodes $v_{a,t}$ for $a \in \mathcal{A}$ and $0 \le t \le L$, rule nodes $u_{r,t}$ for $r \in R$ and $1 \le t \le L$, and only one charged leaf $x_i$ for each $i \in U$. Set

$$v_{a,0} = \bigvee_{i:\ell(i)=a} x_i,$$
$$u_{r,t} = \bigwedge_{a \in B_r} v_{a,t-1}, \qquad v_{a,t} = v_{a,t-1} \vee \bigvee_{r:h_r=a} u_{r,t}. \tag{16}$$

Empty OR is false and empty AND is true. Purchase leaves feed initial atom nodes; no charged copy is introduced at a later layer. Leaves, atom layer zero, rule layer one, atom layer one, and successive rule/atom layers give a topological order even when the original theory has positive cycles.

Induction on $t$ using Equation (1) shows that $v_{a,t}(P) = 1$ exactly when $a \in X_t(P)$. The carry edge implements accumulated closure; a free rule can first affect atom layer one. By Lemma 8, goal $v_{g,L}$ is true exactly for the original supporting subsets. Those subsets and their sums over canonical purchase identities are identical in both models. Restrictions by status and nonnegative repricing consequently preserve the same optimum and witnesses. □

Write $n = |\mathcal{A}|$, $r = |R|$, and $\ell_R = \sum_{r \in R} |B_r|$ after body deduplication. Before goal pruning or binary expansion, the construction has exactly

$$N_U = m + (L+1)n + Lr, \qquad E_U = m + L(n + r + \ell_R). \tag{17}$$

Let $d$ be the number of distinct body atoms. Purchase leaves and rule nodes have fanout one; each such body atom at each layer $t < L$ has a carry consumer and at least one rule consumer. Hence the unpruned shared count is $Ld$. Pruning may reduce it; balancing adds only fanout-one gates.

No small original shared parameter is preserved in general by this unsimplified construction. For a supplied $a_0$ and the unary chain $a_0 \to a_1 \to \cdots \to a_h$ with goal $a_h$, the original DAG has $q = 0$. Here $L = h$. After root-reachability pruning alone, $v_{a_i,t}$ is retained exactly for $0 \le t \le i$: reaching the goal needs at least $h - i$ further rule layers, with any remaining layers filled by carries. For $1 \le i \le h-1$ and $0 \le t < i$, both a carry consumer and a next-rule consumer remain. Thus the pruned unrolling has $\sum_{i=1}^{h-1} i = h(h-1)/2$ shared vertices. Some are constant false and could be removed by further simplification, which is not assumed in this construction. Polynomial unrolling size does not establish tractability in the original graph's $q$.

### B.7 Explicit frontier bounds

An antichain over $m$ acquisitions has at most $K = \binom{m}{\lfloor m/2 \rfloor} \le 2^m$ members, with $K = 1$ at $m = 0$. To see the bound, the prefix sets of any permutation form a chain and meet an antichain $\mathcal{C}$ at most once. Counting permutations by that member gives $\sum_{S \in \mathcal{C}} 1/\binom{m}{|S|} \le 1$, hence $|\mathcal{C}| \le K$.

A direct synchronous implementation can fold each body one atom at a time, minimizing after each union product. Both input frontiers have at most $K$ members. Stream their at most $K^2$ unions into an antichain, comparing each candidate to at most $K$ retained members. Each fold uses $O(K^3)$ mask operations and $O(K)$ extra masks; merging a rule result into its head uses $O(K^2)$ comparisons. With $b = \max(1, m)$ bounding a mask operation's bit cost, conservative bounds are

$$\text{time} = O\big(b\,[m + n + r + \ell_R + L(nK + rK^2 + \ell_R K^3)]\big),$$
$$\text{storage} = O(b\,[nK + m + n + r + \ell_R]). \tag{18}$$

The storage estimate includes old/new atom frontiers, a streamed rule fold, and the input. After compilation, normalizing a query and scanning its goal frontier takes $O((|\mathcal{M}_g| + 1)(m + 1))$ elementary and rational operations. Each cost sum has at most $m$ terms; rational bit complexity is additional.

Exponential frontiers occur in small acyclic inputs. Supply $a_i, b_i$, $1 \le i \le k$, use rules $a_i \to u_i$, $b_i \to u_i$, and $\{u_1, \ldots, u_k\} \to g$. Every minimal support chooses exactly one of $a_i, b_i$ for each $i$, giving $2^k$ supports with $m = 2k$ and $O(k)$ rules and body occurrences. This is a limitation of the explicit frontier, not a lower bound for every implicit solver on this example. All construction bounds refer to an already finite grounded theory; grounding itself may expand a compact input substantially. A stated resource guard may refuse compilation, but must not report that refusal as logical or budget infeasibility.

# C Proofs for Bounded-Price Cache Approximation Bounds

This appendix proves the bounded-price approximation bounds for fixed complete-proof libraries stated in Section 2.4.

This section uses all-available acquisitions and strictly positive bounded prices. Let $\mathcal{F}$ be a finite nonempty family of nonempty supports, $\varnothing \neq \mathcal{L} \subseteq \mathcal{F}$ a fixed complete-proof library, $c(S) = \sum_{i \in S} c_i$, $O(c) = \min_{S \in \mathcal{F}} c(S)$, and $A(c) = \min_{T \in \mathcal{L}} c(T)$. Assume $0 < \underline{c}_i \leq c_i \leq \overline{c}_i$. Empty or zero-cost optima are excluded from this ratio contract. They remain covered by the exact residual-cost theorem. A fixed library is selected before prices are known; assembling new supports after observing prices is a different interface.

## C.1 The classical relative-regret corner formula

*Proof of Equation* (5). The formula is the relative-set-regret characterization of Crema (2018, Lemma 6), with one added to convert regret into stretch. Here is a direct proof under our assumptions. Fix $c$ and choose $S \in \arg\min_{\mathcal{F}} c(S)$. For any $T \in \mathcal{L}$, $d_T(c) = c(T) - c(S) \geq 0$. At the corner $c^S$, costs in $T \setminus S$ increase and costs in $S \setminus T$ decrease, so $d_T(c^S) \geq d_T(c) \geq 0$ and $0 < c^S(S) \leq c(S)$. Thus

$$\frac{c^S(T)}{c^S(S)} = 1 + \frac{d_T(c^S)}{c^S(S)} \geq 1 + \frac{d_T(c)}{c(S)} = \frac{c(T)}{c(S)}.$$

Minimizing over $T$ shows that the expression at this $S$ bounds $A(c)/O(c)$ from above. Conversely, for every $S \in \mathcal{F}$, $O(c^S) \leq c^S(S)$, so $A(c^S)/O(c^S) \geq A(c^S)/c^S(S)$. Taking maxima gives equality. The maximum exists by compactness and positive denominators. The proof does not require every $S$ to be optimal at its own corner, or that all supports have equal cardinality. With positive prices, replacing a feasible superset by a feasible minimal subset cannot worsen its library. □

## C.2 Exact correspondence to covering complete proofs

*All-available case of Theorem 4.* For $g_k = \bigwedge_i (a_i \vee b_i)$, a minimal support contains exactly one acquisition from each pair. Represent it by $x \in \{0,1\}^k$, choosing $a_i$ at zero and $b_i$ at one. At the corner charging one on $x$ and $\kappa$ elsewhere, support $x$ costs $k$, whereas $y$ costs $k + (\kappa - 1)d_H(x, y)$. Equation (5) yields $\Gamma = 1 + (\kappa - 1)R/k$ exactly. A farthest $x$ supplies attaining prices. For $\kappa = 1$ all supports cost $k$, and every nonempty library is exact.

For integer $0 \leq r \leq k$, a library has factor at most $1 + (\kappa - 1)r/k$ if and only if its radius is at most $r$. This proves the identity $N(k, 0, r) = K_2(k, r)$. The ordinary bounds used below are

$$\left\lceil \frac{2^k}{V(k,r)} \right\rceil \leq K_2(k,r) \leq \min\left\{ 2^k, \left\lceil \frac{k2^k}{V(k,r)} \right\rceil \right\}. \tag{19}$$

For any target $1 \leq \gamma \leq \kappa$, the appropriate integer radius is $\lfloor k(\gamma - 1)/(\kappa - 1) \rfloor$. Each radius-$r$ ball covers $V(k, r)$ targets, so covering $2^k$ targets requires at least $\lceil 2^k / V(k, r) \rceil$ centers.

For the upper bound, draw $M = \lceil k2^k / V(k, r) \rceil$ independent uniform centers, allowing repetitions. A fixed target is missed with probability $(1 - V/2^k)^M \leq \exp(-MV/2^k) \leq e^{-k}$. The expected number of uncovered targets is therefore at most $2^k e^{-k} < 1$. Some sample covers the whole cube, and removing duplicates preserves coverage. The full cube supplies the other upper bound. This is a classical probabilistic existence argument; it does not give an efficient optimal-code construction.

To verify the stated asymptotic rate, put $p = r/k$ for $0 < r \leq k/2$. The probability at $r$ is a mode of $\mathrm{Bin}(k, p)$, hence is at least $1/(k + 1)$, giving

$$\frac{2^{kH_2(p)}}{k+1} \leq \binom{k}{r} \leq V(k, r).$$

For every $j \leq r$, $p^j(1-p)^{k-j} \geq p^r(1-p)^{k-r}$. Summing the binomial probabilities up to $r$ gives $V(k, r) \leq 2^{kH_2(p)}$. Equation (19), continuity of $H_2$, and $r = \lfloor \rho k \rfloor$ now prove the limit. At $r = 0$, $K_2(k, 0) = 2^k$ directly. These counting and rate arguments are classical covering theory (Potukuchi & Zhang, 2019); their role here is the exact translation into complete-proof storage and budget distortion. □

### C.3 THE POLYNOMIAL-CACHE THRESHOLD AND THE EASY EXACT CIRCUIT

Let $M = |\mathcal{C}|$. If $R \geq k/2$, the bound $R \geq k/2 - \sqrt{k \ln M/2}$ is immediate. Otherwise put $t = k/2 - R > 0$. For $X \sim \mathrm{Bin}(k, 1/2)$,

$$\mathbb{E}e^{\lambda(k/2-X)} = \cosh(\lambda/2)^k \leq e^{k\lambda^2/8}.$$

Indeed $\log\cosh z \leq z^2/2$ follows by integrating $\tanh z \leq z$ for $z \geq 0$ and using evenness. Markov's inequality at $\lambda = 4t/k$ gives $\Pr(X \leq R) \leq e^{-2t^2/k}$. Counting balls requires $1 \leq M \Pr(X \leq R)$, hence $t^2 \leq k \ln M/2$. Together with $R \geq 0$, the exact radius formula proves the displayed bound on $\Gamma$. If $M_k$ is polynomial in $k$, then $\ln M_k/k \to 0$, and $\liminf_k \Gamma(\mathcal{C}_k) \geq (\kappa+1)/2$ for fixed $\kappa > 1$.

For the antipodal library $\{0^k, 1^k\}$, a target of weight $w$ has distance $\min(w, k-w)$, so $R = \lfloor k/2 \rfloor$. This attains the asymptotic threshold with two proofs. The odd-$k$ finite factor is smaller; no finite equality with the limiting factor is asserted. Any fixed factor below $(\kappa+1)/2$ requires exponentially many complete proofs; the preceding entropy formula specifies the matching exponent for factors strictly between one and this threshold.

In contrast, the exact factored query independently chooses the cheaper acquisition in each pair. It computes $O(c) = \sum_i \min(c(a_i), c(b_i))$ in $O(k)$ arithmetic operations and outputs a witness in $O(k)$. Maintaining this sum permits an $O(1)$ arithmetic update for one changed price. Rational bit costs are additional. Thus the lower bound concerns fixed complete-candidate lookup, not the computational difficulty of the same small circuit or an arbitrary implicit representation.

### C.4 SOUND BUDGET ANSWERS AND FINITE VALIDATION

For any $G \geq \Gamma$, positivity gives $A(c)/G \leq O(c) \leq A(c)$. An affordable cached witness accepts when $A(c) \leq B$; $A(c) > GB$ implies $O(c) > B$ and justifies rejection. Between these tests, the interface is unresolved. The rejection inequality must be strict. For $k = 2$, $\kappa = 3$, antipodal cache, and prices favoring the mixed support $\{b_1, a_2\}$, we have $O = 2$, $A = 4$, $G = 2$: budget two is unresolved by the cache bound despite being feasible; budget one is rejectable, and budget four is acceptable. A factor guarantee is not permission to spend more than the actual requested budget.

The all-available assumption matters. Forbidding $a_1, b_2$ leaves only $\{b_1, a_2\}$ feasible but invalidates both antipodal cached proofs. Bounded prices alone do not preserve feasibility coverage under status changes. This counterexample delineates the price-only result. Appendix D supplies the joint price/status guarantee, while the exact status interface retains its broader contract.

A separate deterministic checker validates these derivations with exact rational arithmetic. On every antichain of nonempty subsets of three acquisitions and every nonempty library, five fixed boxes give 240 library/box cases, 4,368 price-grid queries, and 17,472 sound budget checks. The boxes include equal, fractional, unequal, and fixed coordinates. Direct price evaluation matches the corner maximum and never exceeds it on the midpoint grid. Independently, all 273 nonempty binary libraries at $k \leq 3$, with three fixed $\kappa$ values, give 819 library/price-range cases and 33,417 distinct vertex queries. They match the radius formula. Exhaustive coverage computes every $K_2(k, r)$ at $k \leq 4$; the antipodal radius is checked against every target at $k \leq 12$. The existing Horn compiler agrees with independent truth-subset enumeration at $k \leq 5$. The two budget and availability examples above are checked separately. These are theorem fixtures, with no model, timing, or prevalence claim; the proofs establish the continuum and asymptotic statements.

## D PROOFS FOR PRICE-AND-STATUS ROBUST COMPLETE LIBRARIES

This appendix extends the library analysis to evidence exclusions and proves the joint price/status bounds.

This section proves the joint result in Theorem 4. The goal is $g_k = \bigwedge_{i=1}^{k}(a_i \vee b_i)$, $k \geq 1$, on distinct acquisitions. Each price is in $[1, \kappa]$, rational $\kappa > 1$, and none is acquired. Statuses may exclude one alternative in each of at most $s$ different pairs, $0 \leq s \leq k$, never both alternatives of a pair. Consequently the optimum is always finite and at least $k$. The fixed nonempty library contains complete minimal proofs and may only filter and select them. The full residual-cost interface permits more general states; those are not implicitly included in this ratio theorem.

### D.1 STATUS FACES AND THE EXACT DISTORTION

For $J \subseteq [k]$ and $z \in \{0,1\}^J$, write $F(J,z) = \{x \in \{0,1\}^k : x_J = z\}$. This face is precisely the feasible proof family when the alternative opposite to $z_i$ is excluded at each $i \in J$. Write $\mathcal{C}(J,z) = \mathcal{C} \cap F(J,z)$.

*Exact formula and finite-stretch condition.* Suppose first that every face of codimension $s$ contains a library word. Fix any such face. The positive-box corner formula in Appendix C applies to its feasible family and surviving library. At the corner favoring $x \in F(J,z)$, every item selected by $x$ costs one and every other item costs $\kappa$. The true optimum is $k$, attained by $x$, and any surviving $y$ costs $k + (\kappa - 1)d_H(x,y)$. Taking the minimum over surviving words and then the maximum over faces and targets gives

$$\Gamma_s(\mathcal{C}) = 1 + \frac{\kappa - 1}{k} R_s(\mathcal{C}).$$

Fewer than $s$ exclusions cannot increase this maximum. Given a target $x$ and $|J| < s$, extend $J$ to $J'$ of size $s$ and require $x_{J'}$. The set of matching cached words only shrinks, so its minimum distance cannot decrease. Conversely, size-$s$ masks are allowed statuses, giving equality for the whole declared domain.

If some codimension-$s$ face contains no cached word, that status still has a feasible proof but has infinite cached cost. Its inner radius and the worst ratio are both infinite. Thus finite stretch is equivalent to every assignment appearing on every $s$ coordinates. For $s \geq 1$ this is exactly the classical binary covering-array property (Quistorff & Schlage-Puchta, 2011, Definition 2); for $s = 0$ it is nonemptiness. This use does not identify our constrained radius with the approximate projected matching of generalized surjective codes.

At the separate endpoint $\kappa = 1$, a status-covered library has ratio one, while a missing face still gives infinity. One must not evaluate $0 \cdot \infty$ in the formula. At $s = k$, each face is a singleton: finite stretch requires the full cube, whose radius is zero. □

### D.2 FINITE STORAGE BOUNDS AND THEIR MATCHING EXPONENT

*Proof of Equations* (7) *and* (8). Put $n = k - s$ and $V = V(n,r)$, with $0 \leq r \leq n$ integer and $V(0,0) = 1$. Since $\kappa > 1$, a factor at most $1 + (\kappa - 1)r/k$ is equivalent to $R_s \leq r$. Fix one mask $J$ of size $s$. Its $2^s$ faces partition the cube, and the library words within each face must radius-$r$ cover its $2^n$ targets. Each such word covers $V$ targets. Summing over faces gives $|\mathcal{C}|V \geq 2^k$ and therefore the lower bound. In particular $|\mathcal{C}| \geq 2^s$ is necessary regardless of the radius.

For the upper bound draw $M = \lceil 2k\, 2^k / V \rceil$ independent uniform binary words of length $k$. For any fixed $(x, J)$ with $|J| = s$, a sampled word matches $x$ on $J$ and has distance at most $r$ elsewhere with probability $p = V/2^k$. The probability that all sampled words miss this request is $(1-p)^M \leq e^{-Mp} \leq e^{-2k}$. There are $2^k \binom{k}{s}$ requests, so their expected uncovered count is at most

$$2^k \binom{k}{s} e^{-2k} \leq \exp\big((2\ln 2 - 2)k\big) < 1.$$

Some sample covers every request; removing repetitions preserves coverage. The full cube supplies the other upper bound. This is a probabilistic existence argument, not an efficient construction of a minimum library. The elementary count and union bound suffice to identify the exponential rate.

For $s = \lfloor \alpha k \rfloor$ and $r = \lfloor \rho k \rfloor$ under the stated strict interior conditions, $n/k \to 1 - \alpha$ and $r/n \to \rho/(1-\alpha) < 1/2$. The binomial-volume bounds proved in Appendix C yield

$$\frac{2^{nH_2(r/n)}}{n+1} \leq V(n,r) \leq 2^{nH_2(r/n)}.$$

Taking logarithms of the two storage bounds, their discrepancy is $O(\log k)$. Dividing by $k$ and using continuity of entropy gives Equation (8). At $r = 0$, each target must itself be stored, hence $N(k,s,0) = 2^k$. When $r \geq \lfloor n/2 \rfloor$, $V(n,r) \geq 2^{n-1}$ for $n \geq 1$, so the bounds put $N$ between $2^s$ and $4k2^s$. Thus for a limiting linear fraction of exclusions the storage exponent in this radius range is $\alpha$. At $n = 0$, only $r = 0$ is allowed and the exact value is $2^k$. □

### D.3 THE POLYNOMIAL-STORAGE TRANSITION

Let $M = |\mathcal{C}|$ and suppose $R_s$ is finite, so $M \geq 2^s$. For $n = k - s$, the following nonasymptotic consequence is useful:

$$R_s \geq \max\left\{0, \frac{n}{2} - \sqrt{\frac{n}{2}\ln(M/2^s)}\right\}. \tag{20}$$

For $n = 0$ the radius is zero and the expression is defined directly. If $R_s \geq n/2$ the inequality is immediate. Otherwise let $t = n/2 - R_s > 0$. The counting bound gives $1 \leq (M/2^s)\Pr\{\mathrm{Bin}(n, 1/2) \leq R_s\}$. The moment-generating-function argument of the preceding appendix bounds this tail by $\exp(-2t^2/n)$. Rearrangement gives Equation (20); nonnegativity gives the maximum with zero. The logarithm is used only after establishing $M \geq 2^s$.

Consider a sequence of problem sizes with fixed $\kappa > 1$. If a polynomial-size library has finite stretch for every allowed status, $2^s \leq M \leq k^d$ for some fixed $d$ eventually, so $s = O(\log k)$. Conversely, for $s = O(\log k)$ choose $r = \lfloor (k - s)/2 \rfloor$. Eventually $n \geq 1$, and the upper construction needs at most $4k2^s$ proofs, a polynomial number. Its factor is at most $1 + (\kappa - 1)\lfloor (k - s)/2 \rfloor / k$. For any polynomial library with finite stretch, Equation (20) and $s = O(\log k)$ give

$$\liminf_{k\to\infty} \Gamma_s \geq \frac{\kappa + 1}{2}.$$

The construction attains this limit. Thus the polynomial-storage feasibility transition is exactly $s = O(\log k)$, and the optimal limiting factor within that regime is $(\kappa+1)/2$. Finite exceptions, including $s = k$, are covered by the endpoint result. These statements do not say that two antipodal proofs survive logarithmically many exclusions: those two proofs attain only the all-available threshold.

The factorized circuit independently selects the cheaper available alternative in each pair. It has an exact $O(k)$ arithmetic query and $O(1)$ cost update for one changed pair, with $O(k)$ witness extraction. Excluding both alternatives makes the circuit infeasible immediately, outside the feasible-status ratio definition. This comparison distinguishes fixed whole-proof lookup from retained compositional structure; it is not a hardness result for arbitrary query algorithms.

### D.4 ATTAINING EXAMPLES, SOUND BUDGETS, AND EXACT CHECKS

For $\mathcal{C} = \{0^k, 1^k\}$ and $k \geq 2$, the radii are $R_0 = \lfloor k/2 \rfloor$, $R_1 = k - 1$, and $R_2 = +\infty$. The first is ordinary antipodal coverage. With one fixed coordinate, exactly one antipode remains; a target differing in every other coordinate attains distance $k - 1$. Fixing opposite bits in two coordinates leaves no antipode, proving the last claim.

For $k = 3$, $\kappa = 3$, choose target $x = 011$, force its first coordinate to zero, and charge one on $x$ and three on its alternatives. The optimum $x$ costs three; the only surviving antipode $000$ costs seven. This attains $G = 7/3$. For any price/status query in the declared domain, $A_{\mathcal{C}}/G \leq C_g \leq A_{\mathcal{C}}$. Consequently $A_{\mathcal{C}} \leq B$ accepts, $A_{\mathcal{C}} > GB$ rejects, and all other budgets remain unresolved. At these attaining prices, $B = 3$ is unresolved and $B = 2$ is rejected. Replacing strict rejection by $A_{\mathcal{C}} \geq GB$ would wrongly reject the feasible equality case. A cache with a missing face has no finite factor and cannot turn its empty survivor list into infeasibility.

A standalone exact checker enumerates all 273 nonempty libraries for $k \leq 3$ and 1,071 radius/status cases. For price boxes $[1, 3]$, $[2, 3]$, and $[1, 1]$, a separate surviving-proof cost calculation evaluates 892,701 price/status queries across all vertices and faces. It verifies 1,533 finite-stretch equalities, 1,680 missing-face cases, and all 1,071 constant-price status cases. A distinct request-bitset cover calculation obtains 34 exact minimum-size cases at $k \leq 4$ and agrees with direct radius enumeration where both run. Twelve antipodal examples at $k \in \{2, 3, 4, 8\}$ and $s \in \{0, 1, 2\}$ check coverage loss and the two finite factors; the strict budget boundary is checked separately. The checker is supplied as `submission/check_price_availability_library.py`, with its complete deterministic report. These finite fixtures distinguish the cost and distance formulations and exercise the boundary conventions. They are neither the proof of the general theorem nor external retrieval experiments. The formal allocation case remains unchanged.

# E EXPERIMENTAL PROTOCOL AND REPRODUCIBILITY DETAILS

This appendix records the fixed-case input protocol, independent checks, and retained evidence behind Section 3.

## E.1 INPUT RELEASE AND ALLOCATION INTERPRETATION

The official archive is `proofwriter-dataset-V2020.12.3.zip`, retrieved through the official dataset landing page. Its size is 214,185,889 bytes. Its SHA256 is

```
bbc5694901e8306d0bd659aa1ad53ccfd02c201864f4b320ffa3777827d1fc26
```

The selected member is

```
proofwriter-dataset-V2020.12.3/OWA/depth-3/meta-test.jsonl
```

The first 32 physical lines have 428,260 bytes. Their SHA256 is

```
66d5ee37ea4b3fd5a4567e83ab5fcb1a2645ab72a3e5c6ca3c3e63f0e28cb724
```

Selection does not filter by answer, derivability, support count, price gap, or grammar success. The complete archive hash was checked before the producer read the prefix. The later summary rechecked the same prefix and line identities; it did not replay the entire archive's CRC computation.

Each supplied fact record is a separate acquisition identity, qualified by its source theory identity. Equal literals with different source identities remain alternative purchases. The adapter parses the formal representations of the supplied triples and rules, together with the requested goal. It uses the finite entity constants in those inputs; attribute words are not added as entities. A rule's generic variable is instantiated over that entity domain. Explicit body negation and negative fact/head signs become signed atoms, without closed-world inference. Unsupported grammar is a reported failure. The adapter receives neither a selected gold proof nor a dataset answer.

This is an allocation interpretation of a formal benchmark. The supplied facts are possible purchases, not already-free context; all source rules are free. Prices and acquired/forbidden states are prescribed by this study, not annotations from the dataset. The open-world test member may contain cycles in the union of its rules. Finite closure uses the number of distinct ground rule heads, never the depth of one annotated proof.

## E.2 INDEPENDENT COMPUTATIONS AND CHECKS

For each goal the reference iterates over all $2^m$ subsets of the original fact identities and computes accumulated closure using sets of signed atoms. This produces an independent truth table. Minimal true masks are recovered by subset inclusion and compared with the compiled support family. The reference evaluates each state using subsets containing all acquired facts and containing no identity outside the acquired/available set; additional cost excludes acquired identities. This reference does not call the support-product recurrence to decide truth or cost.

The producer checks exact rational optimum equality, including the absence of a finite optimum. For every returned witness it checks the purchase and selected-set identities, state eligibility, exact additional cost, and forward derivability. Equal optimal costs with different witnesses are accepted and recorded. The direct kernel resolves ties by its compiled identity order. The scalar tree control updates a witness only on strict cost improvement, so a saved choice among scalar ties is not a universal property of all optimal trees.

For a feasible query, write $b_* = s_{\mathrm{paid}} + C$. Tested budgets are $b_*$, $b_* + 1/6$, and $b_* - 1/6$ when $b_* > 0$. All feasible boundaries in this case are at least one, so all three budgets are nonnegative. For an infeasible query, tested budgets are $s_{\mathrm{paid}}$, $s_{\mathrm{paid}} + \sum_{i \in U} c_i$, and one above the latter. Each decision is compared with an independent scan of the reference truth table. A resource guard at $m > 20$ reports refusal separately and was never triggered by this prefix.

Table 3: All exact budget checks. Repeated conditions use the same goals.

| Category | Checks | Expected result |
|---|---|---|
| Finite optimum, $b_* - 1/6$ | 456 | Reject |
| Finite optimum, $b_*$ | 456 | Accept |
| Finite optimum, $b_* + 1/6$ | 456 | Accept |
| No feasible support, finite budget | 4,224 | Reject |
| Total | 5,592 | All agree |

Table 4: Saved wall-time components in seconds. These timers are not isolated performance trials; other local work could run concurrently.

| Component | Calls | Sum | Maximum |
|---|---|---|---|
| Grounding | 466 | 0.144374 | 0.0013909 |
| Support compilation | 466 | 0.039890 | 0.0005173 |
| Goal-frontier lookup | 466 | 0.001635 | 0.0000281 |
| Exhaustive preprocessing | 466 | 109.722486 | 1.9267110 |
| Compiled query and witness | 1,864 | 0.031002 | 0.0001590 |
| Exhaustive query | 1,864 | 7.671932 | 0.2670454 |
| Tree control and reconstruction | 1,864 | 0.398833 | 0.0013820 |

### E.3 Original saved-result summary and labels

The completed producer retains goal reports inside record reports inside a completion report. The summary checks this hierarchy's 501 result JSON files and all ten pinned input/source files. It checks every expected condition, source fact order, price vector, acquired/forbidden set, witness cost, occurrence/union/optimum gap, and saved budget comparison. It does not rerun closure, regenerate support masks, or add records. Support-family mask digests bind the producer's comparisons; a digest alone is not a second independent recomputation of the masks.

Only after the run was complete did the summary read evaluator labels on the identical 32-line prefix. It checked that a true-labelled requested goal was provable, and a false/unknown requested goal was not. The comparison contains 466 labels and 932 saved all-available outcomes across two price vectors. There are zero contradictions. No proof annotation was read for this cross-check and no goal was flipped to its complement. This limited label check is distinct from a three-way natural-language reasoning score.

### E.4 Ties, identity changes, and timing interpretation

The saved kernel and exhaustive reference attain equal costs with different purchase sets in two available-state and two changed-state unit-price queries. The tree union and kernel attain equal costs with different sets in six available-state and three changed-state unit-price queries. Neither comparison has such a difference under the prescribed cyclic prices. These counts do not enumerate all optimal witnesses.

Among goals remaining feasible after the state change, 32 saved purchase sets change under unit prices and 35 under cyclic prices. Repricing changes eight saved optimal sets in the available state and five in the changed state. Witness changes are not automatically cost improvements over the old set, since multiple sets can tie. The theoretical separating-price example establishes a possible failure under the full price domain; these observed changes are only finite-case diagnostics.

The complete producer wall time is 147.311723 seconds. Component timers do not assign every budget scan, check, serialization step, or I/O overhead. Compilation and lookup occur once per goal, serving four queries. The tree control timer includes occurrence and union reconstruction and witness closure checking. The case has small frontiers and was selected for exact verification, so ratios between these timers should not be interpreted as general speedups or a scaling study.

### E.5 Reproduction boundary

The original source includes independent small-instance tests of the direct support kernel, the signed formal-input adapter, and shared-boundary DAG conditioning. Their edge cases include unsupported positive cycles, empty-body rules, duplicate literals with distinct acquisition identities, zero prices, partial availability, repeated leaves, internal sharing, and status changes. The external case exercises the direct Horn-support kernel. It does not exercise a layered DAG implementation on ProofWriter or benchmark the DAG update bound on this dataset.

The separate check of Corollary 6 uses deterministic theorem fixtures. For $k = 2, 3, 4, 5$, exhaustive acquisition subsets reproduce all $2^k$ minimal supports, every support's separating-price optimum, and the strict tree-union gap. At $k = 2$, three fixed price vectors (including zero and fractional prices) and all $3^5$ acquired/available/forbidden assignments give 729 exact status queries, checked against subset enumeration. A fixed $k = 128$ circuit checks the closed-form optimum and eight predetermined price/status updates, including infeasibility and restoration. It also counts state-pair evaluations: two for an update of $a$, and at most $2(3 + \lceil \log_2 k \rceil)$ for a private leaf. This last fixture checks the implementation against the proved identity; it does not enumerate its exponential support family or constitute a runtime scaling experiment.

The revised cache experiment keeps all original records, queries, and budgets. Its new controls reuse only certified initialization conclusions. The portable replay and the separate raw-input verifier are described in Appendix F; they regenerate the revised policy results without retraining, new model calls, or changing the original tree tie rule. The timing table above belongs to the original producer, not this additional verification.

The complete release identity and retrieval location are recorded for local reproduction. Dataset redistribution rights are not inferred from a code repository's license; a distributable artifact should acquire the release from its official source under the applicable terms. Reproducing the mathematical kernels requires no trained model, annotated proof selection, or additional data campaign.

## F Initialization Certificates and Frozen Cache Analysis

The replay fixes the initial unit-price, all-available optimal support for each goal and evaluates the original four resource queries and three budgets per query. This appendix states exactly what each policy retains, proves the additional bounds, and separates the reasons for unresolved decisions.

### F.1 Retained information and sound decisions

Initialization solves the fixed goal exactly with $I = F = \varnothing$, $A = U$, and unit acquisition prices. If no support exists, the initialization can certify that fact; otherwise it returns an optimal support $S_0$ and certifies its minimum cardinality $k_0 = |S_0|$. Merely finding a proof does not certify minimum cardinality, and an incomplete search failure does not supply the infeasibility bit. All four policies below share this initialization; they differ in the conclusions and supports retained or used at query time.

Let $\widehat{C}$ be the current cost of $S_0 \setminus I$ when $S_0$ is eligible, and $+\infty$ otherwise. An absent cached support also gives $\widehat{C} = +\infty$. For any valid total-cost lower bound $L$, the rule is

$$D(B) = \begin{cases} \textsc{accept}, & s_{\mathrm{paid}} + \widehat{C} \le B, \\ \textsc{reject}, & L > B, \\ \textsc{unresolved}, & \text{otherwise.} \end{cases} \tag{21}$$

All budgets here are finite and every comparison uses exact rationals.

**One support with the original bounds.** For an initially provable goal in the all-available state, all prices lie in $[1, 2]$. Every support costs at least $k_0$ and $S_0$ costs at most $2k_0$, so $\widehat{C}/2 \le C_g \le \widehat{C}$. The original lower bound is

$$L_0 = s_{\mathrm{paid}} + \begin{cases} \widehat{C}/2, & I = F = \varnothing,\ A = U,\ \widehat{C} < \infty, \\ 0, & \text{otherwise.} \end{cases} \tag{22}$$

The factor-two argument is not transferred to acquired or excluded states. This policy discards the initial infeasibility conclusion and returns UNRESOLVED when no witness or bound certifies an outcome.

**Adding the initial infeasibility certificate.** Let $z_0$ indicate certified initial infeasibility. If $z_0 = 1$, set $L_{\mathrm{neg}} = +\infty$; otherwise set $L_{\mathrm{neg}} = L_0$. To prove validity, suppose $g \notin \mathrm{Cl}(U)$. Every future eligible $I \cup Q$ is a subset of $U$, and Horn closure is monotone, so $g \notin \mathrm{Cl}(I \cup Q)$. Changes in prices, acquisitions, or eligibility within the same universe cannot create a proof. The certificate concerns the fixed rules, goal, and acquisition universe; adding facts outside that universe or changing the rules changes the query contract.

**Adding the minimum-cardinality bound.** For $z_0 = 0$, put $h = \max\{k_0 - |I|, 0\}$ and sort current available prices as $c_{(1:A)} \le \cdots \le c_{(|A|:A)}$. Define

$$\ell_{\mathrm{card}} = \begin{cases} \sum_{j=1}^{h} c_{(j:A)}, & |A| \ge h, \\ +\infty, & |A| < h, \end{cases} \qquad L_{\mathrm{init}} = \max\{L_0, s_{\mathrm{paid}} + \ell_{\mathrm{card}}\}. \tag{23}$$

For $z_0 = 1$, set $L_{\mathrm{init}} = +\infty$. For any successful completion $Q \subseteq A$, the set $I \cup Q$ is a support and thus has cardinality at least $k_0$. Since $I$ and $Q$ are disjoint, $|Q| \ge k_0 - |I|$. Nonnegative prices imply that any such $Q$ costs at least the sum of the $h$ cheapest available items. If there are fewer than $h$ available items, no successful completion exists. Thus $L_{\mathrm{init}} \le s_{\mathrm{paid}} + C_g$, with the usual extended-real convention. This proof allows irrelevant acquired items; counting them only weakens the bound. Neither this policy nor the preceding one consults the evaluated query's exact optimum or additional proof supports.

**Complete frontier.** Equation (4) provides the exact cost and a minimizing witness, so every finite budget is accepted or rejected. The independent closure reference verifies these answers. This policy differs from a single support in both its alternative witnesses and the completeness information that makes exclusion of all affordable supports conclusive.

### F.2 Coverage and the remaining gap

Figure 3 compares all four policies. Its second panel excludes the 338 initially unprovable goals, exposing the remaining tradeoff on the same 1,536 initially provable budget tests. The original and negative-augmented controls coincide on that subset. Initial minimum cardinality further improves rejection without changing any cached witness.

The infeasibility bit turns 4,056 unresolved cases into correct rejections; the minimum-cardinality bound adds 184. Of the remaining 409 unresolved budgets, 221 are below a finite optimum and 162 have no eligible support at all. These 383 cases require a stronger rejection certificate than the specified bounds supply. The other 26 are feasible and require an affordable witness absent from the cache. For these cases, the binary rule "no affordable cached support implies rejection" is wrong, while all three sound caches return UNRESOLVED. They concern 13 cyclic-price queries over eight goals: 16 budget cases are all-available and ten have changed states. No feasible query in this prefix loses the cached entry through ineligibility alone.

Table 5 separates frontier sizes. Of the 409 unresolved cases, 355 arise from goals with exactly one minimal support. A retained support by itself does not certify that its frontier is complete; the initialization-aware control stores neither that certificate nor an exact state-dependent lower bound. The remaining 54 arise from multiple-support goals: 28 infeasible and 26 feasible budgets. Thus the remaining gap is not entirely the effect of omitted alternative witnesses. These are results for the declared bound policies, not a lower bound on all possible algorithms using the same retained information.

### F.3 Portable replay and independent recomputation

From the artifact root, run `python scripts/replay_cache_policies.py` to regenerate the exact per-budget audit, summary, and result tables. The replay checks every retained goal file

Table 5: All budget outcomes stratified by the initial minimal support family. The four repeated resource queries and three budgets per query give 12 budget tests per goal. Policies add initialization information cumulatively.

| Initial frontier | Policy | Accept | Reject | Unresolved |
|---|---|---|---|---|
| Empty (338 goals) | One support + original bounds | 0 | 0 | 4,056 |
| | + initial infeasibility | 0 | 4,056 | 0 |
| | + minimum-cardinality bound | 0 | 4,056 | 0 |
| | Complete support frontier | 0 | 4,056 | 0 |
| Singleton (117 goals) | One support + original bounds | 824 | 56 | 524 |
| | + initial infeasibility | 824 | 56 | 524 |
| | + minimum-cardinality bound | 824 | 225 | 355 |
| | Complete support frontier | 824 | 580 | 0 |
| Multiple (11 goals) | One support + original bounds | 62 | 1 | 69 |
| | + initial infeasibility | 62 | 1 | 69 |
| | + minimum-cardinality bound | 62 | 16 | 54 |
| | Complete support frontier | 88 | 44 | 0 |

Table 6: Unresolved decisions by resource condition. Each row contains the same 466 goals and 1,398 budget tests. Unit-price, all-available queries are fully resolved once their certified initial minimum cardinality is used.

| Prices | State | One support | + negative | + cardinality | Frontier |
|---|---|---|---|---|---|
| Unit | Available | 1,142 | 128 | 0 | 0 |
| Unit | Changed | 1,170 | 156 | 149 | 0 |
| Cyclic | Available | 1,157 | 143 | 101 | 0 |
| Cyclic | Changed | 1,180 | 166 | 159 | 0 |

against its original SHA256 and reproduces the original cache rule before evaluating the new controls. Run `python scripts/verify_cache_control.py` for a separate raw-input verification. It independently parses and grounds the original 32-line ProofWriter prefix, computes 309,058 distinct fact-subset closures across the 32 theories, and reconstructs all 466 minimal support families. It then checks all 1,864 optima, 5,592 budget labels, policy bounds, and decisions. It imports neither the producer's kernel nor the policy replay implementation. The figure script consumes the regenerated summary; the artifact README gives the exact command. No additional model inference or new benchmark records are used.

### F.4 Original input and result ledger

The 32 theories contain 250 acquisition identities and 214 source rules, giving 466 requested goals. Theories have 1–16 facts and 3–8 source rules; per-goal grounding has 6–57 atoms and 6–32 rules. The uniform distinct-head bound ranges from 4 to 28. Dataset depth three is therefore not used as a global closure bound. Every goal is retained, including unprovable goals. There are no unsupported inputs, skipped cases, or resource refusals.

**Scope of validation.** The original producer completed in 147.31 seconds on one host. The following timers belong to that original execution, not to the revised policy replay. The saved compilation and query timers sum to 0.0399 seconds over 466 compilations and 0.0310 seconds over 1,864 queries, respectively. These are implementation diagnostics, not an isolated speed benchmark or an end-to-end retrieval-latency comparison. The fixed prefix is small, its maximum support frontier is five, and most goals have no support or just one. It is a favorable setting for explicit compilation. The case verifies the scoped formal contract and exhibits its failure mechanisms; it does not establish their prevalence in document retrieval. Appendix E specifies the retained record and independent checks.

Table 7: All four conditions, each with the same 466 requested goals. The gap columns use feasible queries as their denominator. Occurrence overcount ($T > U$) and a suboptimal purchase union ($U > C$) are different failures. Equality cases are retained.

| Prices | State | Feasible | Infeasible | $T > U$ | $T = U$ | $U > C$ | $U = C$ |
|---|---|---|---|---|---|---|---|
| Unit | Available | 128 | 338 | 17 | 111 | 1 | 127 |
| Unit | Changed | 100 | 366 | 5 | 95 | 2 | 98 |
| Cyclic | Available | 128 | 338 | 16 | 112 | 2 | 126 |
| Cyclic | Changed | 100 | 366 | 5 | 95 | 1 | 99 |
| **All queries** | | **456** | **1,408** | **43** | **413** | **6** | **450** |

**State changes and reuse.** The changed state makes 28 previously derivable goals infeasible under each price vector; no unprovable goal becomes feasible. Among the 100 goals feasible in both states, additional cost decreases/ties/increases in 32/68/0 unit-price cases and 32/65/3 cyclic-price cases. Each changed state has 28 zero-additional-cost goals, with past payment still present. Repricing changes eight saved optimal purchase sets in the available state and five in the changed state. Changed witnesses alone do not establish strict suboptimality of the old witness.