Certified Compression of Game-Value Tables

Mantis Discovery

September 2026

Abstract

Exact game-value queries can be recovered from signed distance intervals stored on only part of an arena. We prove a sound and complete endpoint certificate for any retained subset of a finite alternating game with irreversible budget resets. The omitted region may contain cycles. Two extremal boundary problems recover its sharp componentwise bounds without increasing interval width. For widths at most w, certification takes $O((w + \log(N + 2))(N + E))$ graph operations, or $O(w(N + E))$ when the omitted region is acyclic. Under periodic retention, we characterize the exact worst-case refinement depth and prove a matching checkpoint-density bound. Adjacent checkpoint pairs are asymptotically optimal for one-sided storage. A dynamic program also minimizes additive interval-coding cost subject to a prescribed depth bound. A chess implementation using the existing paired grids stores 26 complete pawnless classes in 7,585,632 bytes, 61.64% below their published Syzygy WDL/DTZ inventory. All files pass endpoint certification. The representation preserves all five outcome categories and exact fresh-clock values; its seven-ply quiet-depth bound does not prevent near-threshold p99 latency of 133 milliseconds. The general checker and grid optimizer are tested separately and are not credited with these storage savings.

1 Introduction

A threshold query needs less information than an exact distance. If a winning position has phase distance in $(32, 40]$, a budget of 50 is sufficient and a budget of 20 is not. Only budgets strictly inside the interval require reconstruction. In a game table, legal moves can supply that reconstruction. The difficulty is to bound its cost and certify its answers without consulting the exact table used by the encoder.

We study finite alternating games in which designated moves reset a budget and lower a finite potential. A signed interval records a basic winner and bounds on a min–max phase distance. Together, these determine both unlimited-play outcomes and values under an expiring budget. To certify an interval, the verifier tests its two endpoints, forcing the first move rather than allowing the submitted interval to answer its own test.

The retained vertices may be chosen arbitrarily. Make them absorbing, solve for the omitted region under the smallest and largest submitted boundary ranks, and use the resulting intervals as shortcuts. These bounds are sharp for the conditional boundary problem and do not widen the uncertainty, even through cycles. We prove that this reconstruction preserves the submitted evaluator before its labels have been certified. Endpoint tests then certify the whole arena. When the omitted region is acyclic, as with one-sided storage, the preliminary boundary calculation reduces to a linear-time pass.

Query depth requires a separate argument. If a quiet path reaches a retained vertex every s plies, only checkpoints in the appropriate residue class can stop it. An exact worst-case depth formula gives a matching lower bound on checkpoint density. In particular, adjacent checkpoint pairs are asymptotically optimal for one-sided storage. We then optimize nonuniform grids for an additive coding objective without relaxing their depth bound. This distinguishes a guarantee about reconstruction from a choice of code for the observed ranks.

The chess implementation combines one-sided intervals with arithmetic indexes, reversible block predictors, and certified reconstruction rules. It covers 26 complete pawnless classes and 165,964,170

valid canonical positions. One rule removes an entire 42,590,464-position bishop sector. All dependencies and unfavorable classes remain in the storage comparison. The arbitrary-retention checker and grid optimizer have separate reference implementations; the chess files use paired grids rather than optimized nonuniform grids.

Prior work. Single-sided DTZ, capture reconstruction, offset categories, optional ply-accurate noncursed distances, and tablebase consistency verification are established features of Syzygy [1, 2]. Rule-constrained endgame distances also predate this work [3]. Value compression of pattern databases groups distances into intervals to obtain admissible search estimates [4]. Our decoder instead reconstructs exact adversarial threshold answers. Partial-information endgame databases propagate bounds when some boundary values remain unknown [11]; here all boundaries are certified and every query is resolved exactly. Gomboc et al. compare lossless representations of five-category chess outcomes and their query costs [5]; the present representation also retains phase information.

Tablebase verification has a substantial history. Hurd constructs endgame databases with correctness proofs checked by a theorem prover [12]. A recent preprint by Pavlov studies verification based on capture and quiet moves [13]. Our checker is ordinary executable code, not a proof-assistant formalization. The claims here concern the signed interval certificate, sharp extension through an omitted region, and the exact density–depth tradeoff for its decoder. The phase recurrence is a positive-cost adversarial shortest-path problem [8]; arithmetic indexing and predictive coding remain supporting techniques [6, 7].

2 Basic values and phase distances

Let a finite directed arena have nonterminal vertices owned by $+1$ or -1, with alternating owners on continuing moves. Every nonterminal has a legal move. Terminals have outcomes in $\{-1, 0, +1\}$, and a terminal victory reached by a move belongs to its mover. Some edges reset a budget. A finite-valued potential Φ weakly decreases on continuing edges and strictly decreases on resets. Thus no directed cycle contains a reset.

In the basic game, infinite play draws. Write $u(p)$ for its value, given by the least reachability attractor of each player. Fix a capacity $H \geq 1$. In the clock game Γ_H, a live state (p, r) has reserve $1 \leq r \leq H$. A terminal successor takes its outcome first; a reset to a nonterminal restores reserve H; a quiet move consumes one unit and draws if its nonterminal successor has reserve zero. Every continuing move decreases (Φ, r) lexicographically. Write $V_H(p, r)$ for the value and $w_H(p) = V_H(p, H)$.

A win at reserve r remains a win at any larger reserve: simulate the smaller budget until a reset makes the clocks coincide. To represent both this clock value and the basic value, consider the cost of the first phase, ending at a terminal or reset. For target player σ, a terminal win costs one. A reset into a basic win for σ costs one if its successor is a strict full-reserve win, and $H + 1$ otherwise. An unfavorable exit or failure ever to end the phase has cost infinity.

Let $z_{H,\sigma}$ be the resulting min–max phase distance. A σ-winning terminal has rank zero; other terminals have infinite rank. The edge requirements are

$$b_\sigma(p, q) = \begin{cases} 1, & q \text{ is a terminal } \sigma\text{-win}, \\ \infty, & q \text{ is another terminal}, \\ 1 + H\mathbf{1}\{w_H(q) \neq \sigma\}, & (p, q) \text{ resets and } u(q) = \sigma, \\ \infty, & (p, q) \text{ resets and } u(q) \neq \sigma, \\ 1 + z_{H,\sigma}(q), & (p, q) \text{ is quiet and } q \text{ is nonterminal}. \end{cases} \tag{1}$$

At a nonterminal take the minimum when $\mathrm{own}(p) = \sigma$ and the maximum otherwise. Finite distances refer to strategies that force a phase exit; a positive-cost cycle does not receive a finite rank.

Proposition 2.1 (Signed phase representation)**.** *A nonterminal has finite $z_{H,\sigma}(p)$ exactly when $u(p) = \sigma$. On basic decisive positions write $z_H(p) = z_{H,u(p)}(p)$. Then*

$$V_H(p, r) = \begin{cases} u(p), & u(p) \neq 0 \text{ and } z_H(p) \leq r, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \leq r \leq H. \tag{2}$$

The five-category value from the mover's perspective is

$$W_5(p) = \begin{cases} 0, & u(p) = 0, \\ 2u(p)\,\mathrm{own}(p), & z_H(p) \leq H, \\ u(p)\,\mathrm{own}(p), & z_H(p) > H. \end{cases} \tag{3}$$

A positional basic winning policy decreases z_H on quiet continuations and Φ on resets. If the entering reserve covers z_H, it also wins the clock game.

Proof. A basic attractor policy reaches a winning terminal or a reset into a lower basic win without repeating a vertex. Its worst-case phase cost is finite. Conversely, a finite phase strategy reaches such an exit; composing the lower-stratum winning strategies proves a basic win. Positive-cost min–max propagation determines these proper phase values [8].

For $r \leq H$, a phase of cost at most r must end within r moves at a winning terminal or a reset into a strict full-reserve win. The $H + 1$ toll cannot be paid. These are exactly the ways a clock win can end its first phase. Induction over reset strata and the remaining reserve proves (2), hence (3). Choosing minimizing edges in (1) decreases the finite rank on every quiet continuation, including all opposing replies. Resets lower Φ; strict wins reset only into strict full-reserve wins. The policy therefore makes progress and preserves its stated outcome. □

The values $+1$ and -1 in (3) are a cursed win and a blessed loss: a basic decisive outcome drawn by the clock rule. The offset distinguishes an overlong present phase from a short phase leading to a later clock obstruction. This convention is related to native DTZ, but the represented integer is the exact cost in (1), not a particular release's rounded value or total distance to mate [2, 3].

Only the basic winner and strict full-reserve status of a reset successor are needed in (1); its exact phase distance is unnecessary. Call a layer closed under quiet moves if every nonterminal quiet successor stays in it. With n nonterminals and only terminal or solved reset exits, its finite ranks are at most $H + n$: an attractor policy uses at most $n - 1$ preceding quiet moves and an exit of cost at most $H + 1$. This supplies a finite rank cap, not a one-byte rank bound.

3 Partial tables and endpoint certificates

Work in a layer closed under quiet moves, with N nonterminals and E legal edges. Every exit is terminal or enters a certified lower layer, so its basic outcome and the phase costs in (1) are known. Fix any set S of retained vertices and put $D = V \setminus S$. The omitted subgraph may contain cycles.

A retained nonterminal carries a draw symbol or a signed cell

$$(\sigma, L(p), U(p)), \qquad \sigma \in \{-1, +1\}, \quad 0 \leq L(p) < U(p). \tag{4}$$

Its intended meaning is $u(p) = \sigma$ and $L(p) < z_H(p) \leq U(p)$. The verifier first checks domain coverage, terminal handling, the reset grading, and the syntax of every cell. The endpoints are not yet assumed correct.

To reconstruct proposed signs on D, temporarily make each retained vertex absorbing, with the outcome written in its label. Solve the two reachability games on D with these boundary outcomes and the certified exits. Infinite containment in D draws. Write $\widehat{u}$ for the resulting signs on D together with the submitted signs on S. The ordinary basic Bellman equation holds on D by construction; check it at every vertex of S as well. At a $+1$ vertex it takes the maximum successor sign, and at a -1 vertex the minimum.

3.1 Sharp intervals on the omitted region

The same boundary problem recovers numerical bounds. For each decisive retained vertex q, choose an integer boundary rank

$$\theta(q) \in \{L(q) + 1, \dots, U(q)\}.$$

For a target σ, a quiet move into a retained σ vertex q has cost $1 + \theta(q)$; another retained sign is unfavorable. Original exits keep their certified costs. Let $F_\theta(p)$ be the proper min–max cost on an omitted vertex with proposed winner σ. Failure to reach a favorable boundary or exit has infinite cost. Its winning region depends on the boundary signs, not on the finite numbers θ.

Set $\theta^-(q) = L(q) + 1$ and $\theta^+(q) = U(q)$, and define on decisive omitted vertices

$$\ell(p) = F_{\theta^-}(p), \qquad v(p) = F_{\theta^+}(p), \qquad (L(p), U(p)] = (\ell(p) - 1, v(p)]. \tag{5}$$

The symbol v in this equation is a numerical upper bound, distinct from the game value u.

Theorem 3.1 (Extremal boundary extension)**.** *For fixed proposed boundary signs, F_θ is monotone in θ and is nonexpansive in the supremum norm on its finite winning region:*

$$\|F_\theta - F_{\theta'}\|_\infty \le \|\theta - \theta'\|_\infty. \tag{6}$$

For each decisive omitted vertex, the possible integer values under independent boundary choices are exactly

$$\{F_\theta(p) : \theta^- \le \theta \le \theta^+\} = \{\ell(p), \ell(p) + 1, \dots, v(p)\}. \tag{7}$$

Consequently, retained widths at most $w \ge 1$ give omitted widths at most w, even when D contains cycles. With no decisive retained boundary, the omitted bounds are exact.

Proof. Fix a strategy that forces the first visit to a favorable retained vertex or original exit. Each resulting play has a quiet path cost followed by exactly one boundary cost. Raising the boundary ranks cannot lower this cost. Changing every boundary rank by at most ε changes every finite play cost by at most ε; an original exit cost does not change at all. The set of strategies that force a favorable first exit is unchanged. Taking the worst opposing play and then the best forcing strategy preserves monotonicity and the same additive bound. This proves (6) and the extremal bounds in (7).

Both extrema are attained by the indicated boundary vectors. Move from θ^- to θ^+ by increasing one integer coordinate at a time. Monotonicity and (6) make each scalar value increase by either zero or one, so every integer between the extrema is attained. Finally,

$$v(p) - \ell(p) \le \max_{q \in S : \widehat{u}(q) \ne 0} (U(q) - L(q) - 1) \le w - 1.$$

Thus $U(p) - L(p) = v(p) - \ell(p) + 1 \le w$. If there is no decisive boundary, no terminal cost is uncertain and the two problems coincide. □

These are the best componentwise bounds for the absorbing boundary problem with independent boundary ranks. Additional relations imposed by the retained vertices' own moves may tighten them; those relations have not yet been used. The theorem does not assume that arbitrary boundary assignments extend to a correct labeling of the whole arena.

The two extremal problems for each player can be solved by a standard label-setting method for positive-cost min–max paths [14, 15]. A winner's vertex becomes a candidate when one favorable action is resolved; an opposing vertex becomes a candidate only after every action is resolved, using their maximum cost. A priority queue settles vertices in increasing candidate order. Settling a quiet successor at value t supplies an action of cost $t + 1$ to its predecessors. Certified exits and retained boundary costs initialize the queue.

For completeness, each candidate is a genuine upper bound: it has a finite strategy proof using settled children. Among vertices with finite true value that are not settled, take one of least true value. At a minimizing vertex, an optimal quiet child has smaller true value; at a maximizing vertex, every quiet child does. Those children have already settled, or the relevant action is an initialized exit. The vertex therefore has a candidate equal to its true value. The next queue minimum is exact. Induction settles every finite value, leaving precisely the nonwinning vertices unresolved. With an indexed binary heap, each edge is processed once per pass and the four passes take $O((N + E)\log(N + 2))$ operations and $O(N + E)$ storage. This is an application of label setting, not a new shortest-path algorithm.

If D is acyclic, reverse topological evaluation suffices in $O(N + E)$ work. A quiet edge shifts a child's interval by one, an exit of cost b contributes $(b - 1, b]$, and componentwise minima or maxima give (5). In particular, retaining one owner has this cheaper reconstruction.

3.2 The submitted evaluator

For target σ, let $\mathsf{P}_\sigma(p, R)$ evaluate the proposed table at an integer budget $R \geq 0$. A terminal answers from its outcome, including at zero. A nonterminal at zero, or with proposed sign different from σ, answers false. At a retained vertex use

$$R \leq L(p) \implies \text{false}, \qquad R \geq U(p) \implies \text{true}. \tag{8}$$

Otherwise expand legal moves, taking disjunction at a σ vertex and conjunction at its opponent. A quiet nonterminal successor is evaluated at $R - 1$; an exit of cost b_σ answers $b_\sigma \leq R$. A reset to q therefore succeeds precisely when

$$u(q) = \sigma \quad \text{and} \quad \big(R \geq H + 1 \text{ or } \mathsf{P}_\sigma(q, H)\big), \tag{9}$$

where lower-layer predicates have already been certified. Same-layer recursive calls lower R, so even an incorrect submitted table defines a terminating evaluator.

Lemma 3.2 (Safe derived shortcuts)**.** *At omitted vertices, the bounds in* (5) *may also be used in* (8)*. They preserve every answer of the submitted evaluator before the retained cells are certified.*

Proof. If $R \geq v(p)$, a strategy for the upper boundary problem reaches an original successful exit within its cost, or a retained q with remaining budget at least $U(q)$. Its leaves are true shortcuts of the submitted evaluator, so literal expansion is true.

Conversely, a successful literal expansion gives a finite AND/OR proof. Stop each branch at its first retained visit. A true retained predicate at remaining budget r requires $r > L(q)$, whether it used a shortcut or further expansion. Replacing that suffix by the optimistic boundary cost $L(q) + 1$ can only reduce its cost. The truncated proof is a feasible strategy in the lower boundary problem. Hence $\ell(p) \leq R$. Therefore $R \leq \ell(p) - 1 = L(p)$ must answer false. Replacing the derived shortcuts by these finite arguments preserves all recursive answers. □

A proposed endpoint cannot certify itself. Define $\mathsf{E}_\sigma(p, R)$ by the same procedure, except that the first positive-budget nonterminal call bypasses its shortcut and expands a legal move. For every decisive retained cell require

$$\mathsf{E}_\sigma(p, L(p)) = \text{false}, \qquad \mathsf{E}_\sigma(p, U(p)) = \text{true}. \tag{10}$$

Draw labels have no numerical endpoint tests.

Theorem 3.3 (Certification with arbitrary retention)**.** *For any retained subset S, the reconstructed boundary signs, the sign equations on S, and* (10) *accept precisely the correct retained signs and intervals. Acceptance determines all omitted signs and threshold predicates exactly. No exact ranks on S or D are supplied to the verifier.*

Proof. By Lemma 3.2, use the literal evaluator in the soundness argument. A successful upper test at $p \in S$ is a finite AND/OR proof with leaves at certified exits or true retained shortcuts. A same-layer shortcut at q reached after $j \geq 1$ quiet moves satisfies

$$U(q) \leq U(p) - j < U(p).$$

Induction on the positive integer $U(p)$ replaces each such leaf by a genuine phase-winning strategy of cost at most $U(q)$. The composed strategy proves every decisive retained sign and its upper bound. Each decisive omitted sign is then sound: its boundary-game strategy forces a favorable first exit or retained vertex, after which the corresponding sound strategy applies.

All proposed draw signs are also correct. Bellman consistency gives each player a nonlosing policy on the zero region. On its own turns it chooses a zero successor. If its opponent leaves the zero region, it can enter only a region favorable to that player, whose sign is already sound. Infinite containment draws, and the original exits are certified. Thus both players avoid losing from every proposed draw, so $\widehat{u} = u$ everywhere.

Suppose a retained lower endpoint is false, with $L(p) \geq z_H(p)$. Choose such a vertex with least true phase rank. Follow a true phase-optimal strategy in the failed lower test, choosing a branch responsible for failure at an opposing vertex. Every quiet move strictly lowers the true rank, while the remaining budget still covers it. A correct exit cannot fail and the budget cannot expire first. The false answer must therefore reach a retained shortcut at some q, with remaining budget R such that

$$z_H(q) \leq R \leq L(q), \qquad z_H(q) < z_H(p).$$

This contradicts minimality. All retained cells are correct, and induction on R makes every predicate the true threshold recurrence, including on D.

Conversely, assume the retained cells are correct. A genuine winning strategy on D reaches a favorable original exit or a retained vertex with the same basic winner. A boundary-winning strategy composes with the true retained strategy. Hence boundary sign reconstruction is exact in both directions. The sign equations hold, literal expansion evaluates the true threshold predicate, and forcing its first move gives the correct endpoint answers. Lemma 3.2 preserves them, proving completeness. □

Bellman equations alone can sustain a fictitious winner around a quiet cycle. Upper tests exclude it through strict progress; lower tests exclude a shifted threshold inside a genuine winning region. Taking $S = \varnothing$ is permitted, but then reconstruction solves the entire layer. Arbitrary retention is a correctness result, not a uniform small-query-cost guarantee.

3.3 Verification work

After extending the intervals, put $a_p = U(p) - L(p) - 1$ at decisive vertices and $a_p = 0$ at draws. These count the unresolved integer budgets at each vertex.

Theorem 3.4 (Verification work)**.** *After $O((N + E)\log(N + 2))$ boundary-extension work, all sign and forced endpoint tests take*

$$O\left(N + E + \sum_p a_p(1 + \deg p)\right) \tag{11}$$

local graph operations, with at most $\sum_p a_p$ memoized Boolean predicates. If D is acyclic, the preliminary work is $O(N + E)$. Widths at most w therefore give $O(w(N + E))$ total work in the acyclic case, and $O((w + \log(N + 2))(N + E))$ in general.

Proof. Use the derived shortcuts at every vertex and memoize only unresolved pairs (p, R). There are a_p such pairs at p, each expanding at most $\deg p$ edges once. Their dependencies are acyclic because quiet edges lower R and exits have known costs. The two forced root expansions per decisive retained

vertex add at most $2\sum_{p\in S}\deg p$ edge visits; sign checks add a linear scan. Lemma 3.2 preserves all test answers. Theorem 3.1 gives $a_p \le w-1$, and the stated boundary algorithms give the preliminary cost. □

These are graph-operation bounds, independent of the numerical rank cap. Endpoint arithmetic still uses $O(\log(Z+N+2))$ bits when submitted endpoints and exit costs are bounded by Z. Input access, reverse edges for label setting, signs, endpoints, and lower-layer caches are additional storage. Two bits distinguish an uncomputed, false, or true memoized predicate.

Corollary 3.5 (Rule-generated labels)**.** *A deterministic, terminating procedure may supply any subset of retained labels in place of stored entries. If the verifier regenerates those labels and includes them in the sign and endpoint tests, acceptance certifies the combined representation.*

Proof. The procedure's outputs are ordinary proposed cells. Theorem 3.3 applies to their union with the stored cells. □

The rule need not be trusted as a theorem about values. Its domain, termination, indexing, and access restrictions must still be checked; in particular, it cannot read an undeclared exact table to reconstruct an erased entry.

4 The optimal checkpoint density

Certification allows arbitrary retained sets. A uniform query-depth guarantee needs more structure. Suppose a map $c : V \to \mathbb{Z}/s\mathbb{Z}$ satisfies $c(q) = c(p)+1$ on each quiet nonterminal edge, and retain precisely $c^{-1}(0)$. Starting at a retained vertex, another retained vertex is encountered every s quiet plies unless the phase ends sooner. For $s = 2$, the two owner classes provide this map.

Fix an integer rank cap Z and a common grid $G \subseteq \{0,\dots,Z\}$ containing 0 and Z. Each retained rank lies in the cell between two consecutive elements of G. For queries with $1 \le R \le Z$, define

$$\delta_s(G;Z) = \max_{1\le R\le Z} \min\left(\{R\} \cup \{R-g : g\in G,\ g\le R,\ R-g\equiv 0 \pmod s\}\right). \tag{12}$$

The first choice is budget exhaustion. The others are the distances to earlier checkpoints that will coincide with a retained visit. We count quiet edges traversed before an answer or phase exit; lower-layer work after a reset is separate.

Theorem 4.1 (Sharp refinement depth)**.** *For the interval-shortcut evaluator on this retention pattern, $\delta_s(G;Z)$ is the exact worst-case quiet depth from a retained vertex, over all compatible finite arenas and all correct cells of rank at most Z. A query starting j quiet plies before its next retained visit has depth at most $j+\delta_s(G;Z)$, where $0 \le j < s$. If different blocks use grids containing G, these upper bounds still hold.*

Proof. Along a continuing branch, budgets at retained vertices are $R, R-s, R-2s, \dots$. Every grid endpoint decides a cell: it is either at or below its lower endpoint, or at or above its upper endpoint. Thus the branch stops by the first eligible checkpoint or by budget exhaustion, giving (12). A nonretained start adds at most j moves. Additional checkpoints can only stop the branch earlier.

For sharpness, choose a budget R attaining the maximum. If the maximum is zero there is nothing to prove. Otherwise $R < Z$, since $Z \in G$. Use a forced alternating path whose winning terminal is $R+1$ moves from the retained root. Its basic winner is the mover of the last edge, and its exact ranks are $R+1, R, \dots, 1$. Query the root at budget R. After d quiet moves, the rank is one greater than the remaining budget $R-d$. Whenever this budget is positive and is not a grid endpoint, the two consecutive integers lie in the same cell $(L,U]$, with $L < R-d < U$. The visit is ambiguous. At the first eligible checkpoint the budget equals a lower endpoint and answers false; otherwise it expires at a nonterminal.

All traversed edges are quiet and all ranks are at most Z. The depth is precisely the inner minimum in (12). □

This theorem concerns the ordinary interval decoder, without extra rules, precomputed derived shortcuts on the retained set, or advance knowledge of the queried path. Such information may improve particular instances. The result is not a lower bound on all possible encodings of game values.

4.1 A matching counting bound

For $a \in \{1, \dots, s\}$, let

$$n_a = \#\{r \in \{1, \dots, Z\} : r \equiv a \pmod{s}\}.$$

The representative $a = s$ denotes residue zero. Let $a_Z = 1 + ((Z-1) \bmod s)$ be the positive representative of the cap's residue. The next theorem gives the exact minimum number of positive checkpoints, including the required cap.

Theorem 4.2 (Checkpoint density)**.** *Fix $k \geq 1$ and require $\delta_s(G; Z) \leq s(k-1)$. The minimum number of positive endpoints is*

$$\min_G |G \cap \{1, \dots, Z\}| = \sum_{a=1}^{s} \left\lfloor \frac{n_a}{k} \right\rfloor + \mathbf{1}\{k \nmid n_{a_Z}\}. \tag{13}$$

An optimal grid consists of 0, Z, and

$$a + s(jk - 1), \qquad 1 \leq a \leq s, \quad 1 \leq j \leq \left\lfloor \frac{n_a}{k} \right\rfloor. \tag{14}$$

Consequently the minimum asymptotic endpoint density is $1/k$, or $s/(D+s)$ when the depth bound is $D = s(k-1)$.

Proof. Partition the positive budgets of each residue into consecutive disjoint blocks of k. Let R be the final budget of a full block. It exceeds $s(k-1)$, so exhaustion cannot meet the required depth. A checkpoint of the same residue must lie between $R - s(k-1)$ and R, exactly the block under consideration. Distinct blocks require distinct positive checkpoints. If $k \nmid n_{a_Z}$, the required cap lies outside all full blocks, so it contributes one further endpoint. This proves the lower bound in (13).

For the construction, choose the last budget of every full block in every residue class. Before the first chosen checkpoint of a residue, each budget is at most $s(k-1)$ and can expire within the bound. Thereafter, the distance to the previous checkpoint of its residue is at most $s(k-1)$, including the final incomplete block. Adjoining Z cannot increase depth. Formula (14) selects one checkpoint per full block. The cap is already selected exactly when $k \mid n_{a_Z}$, so the construction attains (13). Since $\sum_a n_a = Z$, dividing by Z and letting Z grow proves the density statement. □

For one-sided storage, put $s = 2$ and $B = 2k$. Formula (14) becomes the pairs $jB - 1, jB$. Thus two adjacent checkpoints every B budgets attain depth $B - 2$ from the stored side with asymptotically minimum checkpoint density. This explains the pairing without assuming a particular rank distribution.

Corollary 4.3 (Paired and nested grids)**.** *For an even $B \geq 2$, let*

$$G_B = \{0, H-1, H, Z-1, Z\} \cup \{jB - 1, jB : 1 \leq jB \leq Z\}, \tag{15}$$

clipped to $[0, Z]$. Correct one-sided cells on this grid give exact basic outcomes, all five outcome categories, and every fresh-clock value. Their quiet depth is at most $B-2$ from the stored side and $B-1$ from the other side. Forced endpoint tests have depth at most B. The same bounds hold for blockwise choices among the nested grids $G_1, G_2, G_4, \dots, G_B$, where B is a power of two and G_1 contains every integer.

Proof. The grid contains the construction of Theorem 4.2 for $s = 2$; the extra endpoint $Z - 1$ covers a possible incomplete final pair. Theorem 4.1 proves the ordinary depth bounds. A forced test first expands from the stored side and, after at most two quiet moves, either ends or returns to it, giving $2 + (B - 2) = B$. Every finer grid in the stated family contains G_B. Exact threshold evaluation and Proposition 2.1 give the values. The checkpoints H and $H - 1$ decide full-reserve queries immediately at the stored side and after at most one quiet move at the other side. □

With exact one-sided labels, ordinary queries need at most one quiet move and forced tests need at most two. Binary search recovers a finite phase rank with at most $\lceil \log_2 Z \rceil$ threshold calls. The representation is therefore exact even when its stored numerical labels are intervals.

A width bound alone gives no comparable query-depth guarantee. Use only checkpoints $0, 4, 8, \ldots, 100$ on a forced 100-ply win, queried at reserve 99 from the stored side. Every retained visit has an odd remaining budget strictly inside its cell, so the search follows 99 quiet plies before exhaustion. All cells have width four. Independent nonnested grids can fail for the same reason: no common residue checkpoint is guaranteed along a path.

Depth does not control the size of a branching search. With degree bound Δ, a direct depth-d search may examine $1 + \Delta + \cdots + \Delta^d$ vertices. Resets add lower-layer queries. The practical choice $B = 8$ bounds a quiet segment by seven plies, not by seven inexpensive operations.

5 Choosing a grid for the observed ranks

Uniform pairs minimize asymptotic checkpoint count, but two grids with the same count may have different coding costs. Let $c(a, b)$ be a specified cost for the cell $(a, b]$, and let Q be a set of required endpoints containing 0 and Z. We seek a grid minimizing

$$\sum_{(a,b) \text{ consecutive in } G} c(a, b), \qquad Q \subseteq G, \qquad \delta_2(G; Z) \le D. \tag{16}$$

This allows a nonuniform grid while retaining an exact worst-case bound.

Theorem 5.1 (Optimal additive-cost grid)**.** *Suppose $c(a, b)$ is available in constant time and its values can be added and compared exactly. Problem* (16) *is solvable in $O(Z(D+2)^2)$ arithmetic operations and $O((D + 2)^2)$ working cost storage. Recovering a grid by retaining all backpointers uses $O(Z(D + 2)^2)$ additional storage. The cost need not be monotone or nonnegative.*

Proof. Scan potential endpoints $t = 1, \ldots, Z$. A state (e, o) records the most recent even and odd checkpoints. Initially $(e, o) = (0, \perp)$, with $\perp$ meaning that no odd checkpoint has been chosen. The most recent endpoint of either parity is $m = \max(e, o)$, ignoring $\perp$.

At step t, either skip t or select it. Selection adds $c(m, t)$ and replaces the checkpoint of t's parity by t. Skipping is forbidden for $t \in Q$. After the choice, if $t > D$, require

$$\text{the most recent checkpoint of parity } t \text{ exists and is at least } t - D. \tag{17}$$

For $t \le D$, exhaustion already meets the depth bound. Equation (12) shows that (17) at every scan step is exactly the feasibility condition. Future endpoints cannot repair a failed condition at an earlier budget.

Keep the least accumulated cost at each state. All completed cells have already been charged, and the last endpoint is known from (e, o). Every possible future cost and every remaining feasibility condition therefore depend only on this state. Induction on t proves the dynamic-programming optimality principle. Selecting Z closes the final cell, so the least terminal-state cost is the optimum in (16).

After the initial segment, the most recent checkpoint of the current parity has age at most D; that of the other parity has age at most $D + 1$, by the preceding step's condition. Before that point there are only $O(D + 2)$ possible values for either component, including $\perp$. Thus each layer has $O((D + 2)^2)$ states

and two transitions per state. Rolling cost arrays and, optionally, a backpointer for every state give the stated bounds. □

One useful objective is an ideal zero-order code length. Let $h_\sigma(r)$ count retained decisive entries of sign σ and rank r, and put

$$n_\sigma(a,b) = \sum_{a<r\le b} h_\sigma(r), \qquad M = \sum_{\sigma,r} h_\sigma(r).$$

For $M > 0$, a per-cell description charge λ gives

$$c(a,b) = \lambda + \sum_{\sigma\in\{-1,+1\}} n_\sigma(a,b)\log_2 \frac{M}{n_\sigma(a,b)}, \tag{18}$$

where a zero summand is omitted. Prefix sums make the counts available in constant time. Fixed draw costs can be added separately. The objective is the ideal coding cost of the signed cell alphabet conditional on a decisive entry; its description charge must include whichever grid metadata the chosen code requires.

The theorem guarantees an optimum for the stated additive objective, not for arbitrary compressed-frame lengths. Zstandard depends on ordering and cross-symbol structure that (18) does not model. Floating-point evaluation also requires care: the reference entropy examples use binary64 arithmetic, whereas the exact optimality tests use integer costs. Neither the current chess format nor its measured file sizes use grids chosen by this optimizer.

6 Removing entries and coding the remainder

6.1 Reconstruction before lookup

The chess decoder retains one owner and uses the paired grids of Corollary 4.3. Before accessing a stored label, it tests any applicable reconstruction rule. A rule must have a decidable domain and must terminate without reading its erased entries. Corollary 3.5 certifies its output together with the remaining table. Erased labels may then be filled arbitrarily or removed from the index, provided queries cannot reach those locations before the rule is applied.

The elementary rules detect current terminals, captures to strict mover-winning lower boundaries, and positions where every legal move captures. A strict winning capture gives rank one. When every action captures, lower outcomes determine the basic winner and the corresponding phase toll; the general rule retains $H + 1$ when required. A drawing capture alone is insufficient, since a quiet alternative may win. These rules are instances of established capture reconstruction [1, 5].

6.2 An entire bishop sector

In KBBvKN, with White holding both bishops, quiet play preserves whether the bishops occupy the same square colour. On the same-colour sector consider this procedure: test current terminals, then assign each capture its certified lower outcome, each terminal move its actual outcome, and every other quiet move draw. Take minimax over the moves. A decisive nonterminal result receives phase rank one.

Proposition 6.1 (Computed sector certificate)**.** *On the stated KBBvKN arena, the same-colour sector contains 42,590,464 valid canonical positions. Complete checks certify the preceding procedure on all of them. Only 206 are decisive, and every decisive rank is at most one. The sector can therefore be removed from the stored index.*

Proof. The standalone checker certifies the required lower signed files, constructs candidate exact cells from the one-ply procedure, and verifies terminal status, basic signs, and exact finite-phase equations

throughout the sector. These checks imply the exact-cell case of Theorem 3.3: finite ranks supply progress, while consistent draws give both players nonloss. No five-man input or producer rank file is opened. A separate comparison with the audited exact witness agrees at all 42,590,464 positions. The proposition is a finite computational result relative to the move model and checking code. □

The procedure stores neither this sector nor its 206 decisive cases. Moves reconstruct the exceptions as well as the draws. Quiet play cannot enter or leave the sector, so its one-ply rule does not accumulate after a seven-ply search in the retained sector. Elementary capture rules end the phase. The measured configuration consequently retains the seven-ply quiet bound; lower-class recursion and move generation remain additional costs.

6.3 Arithmetic indexing

The pawnless index first chooses one of 462 ordered king-pair representatives under board geometry. Equal nonking units are sorted, and stabilizer duplicates are invalid slots. The represented population is unchanged by the following layouts.

Remove the king squares from the ordered board. Three identical knights then have coordinates $0 \le a < b < c < 62$ and colex index

$$a + \binom{b}{2} + \binom{c}{3} \in \{0, \dots, \binom{62}{3} - 1\}. \tag{19}$$

This reduces the two-sided KNNNvK index from 242,221,056 to 34,945,680 slots while retaining all 30,261,917 valid states. The index and its inverse require no stored permutation map.

For KBBvKN, after removing the kings let n_0, n_1 count available squares of each colour. An opposite-colour bishop pair has index $an_1 + b$, using its within-colour coordinates. Since $n_0 + n_1 = 62$ and $n_0 n_1 \le 961$, the retained stored-side array has padded shape

$$(462, 961, 60),$$

with the last coordinate locating the knight among the remaining squares. It represents all 42,822,220 valid opposite-colour states across the two movers; the complementary sector is reconstructed. Axis order and stored owner are recorded in the file. Compactness alone does not guarantee better compression, because removing invalid slots can also remove useful fill freedom.

6.4 Block completion and reversible predictors

Each block contains at most 16,384 symbols. The encoder chooses a permitted nested grid and fills invalid or reconstructible slots, which are never queried as labels. The small-class menu includes forward, backward, modal, and columnwise fills; five-man encoding uses forward and modal fills. The selected completion minimizes the complete compressed frame length, with deterministic ties.

A second pass tries identity, bit transposition, modular differences of lag one or 64, a 64-column modular predictor, and one- or two-dimensional XOR residuals. For example,

$$e_i = x_i \operatorname{xor} x_{i-1}, \qquad x_{-1} = 0,$$

is inverted by prefix XOR. In two dimensions,

$$e_{r,c} = x_{r,c} \operatorname{xor} x_{r-1,c} \operatorname{xor} x_{r,c-1} \operatorname{xor} x_{r-1,c-1},$$

with zero outside the block; prefix XOR along both axes recovers the input. Modular predictors use analogous prefix sums. These standard reversible filters preserve the completed symbols and checkpoint widths [6, 7]. Every selected frame is inverted and compared with its whole source block, including fill positions.

The filter menu includes identity and uses a fixed-length descriptor. Selecting the shortest complete frame therefore cannot enlarge a block within this format. It gives no dominance guarantee over another format.

Files use a 128-byte header, a 16-byte directory entry per nonconstant block, and independent Zstandard frames. A constant file needs the header and one symbol. Descriptors specify material, owner, dimensions, index mode, axis order, grid, symbol width, reconstruction rules, and source digest. Exact five-man sources use signed two-byte integers; their coarse interval alphabets fit one byte under explicit rank caps above 100. This narrowing is checked on every completed symbol, rather than truncating ranks. There is no separate WDL file, exception list, rule-domain bitmap, or exact-rank fallback.

7 Validation and chess experiments

7.1 Finite tests of the general results

Two Python implementations exercise the certificate independently of the chess code. A budget-by-budget solver supplies exact outcomes and thresholds for small arenas. One checker eliminates an acyclic omitted region; the other uses an indexed priority queue and permits arbitrary retention. Both literal expansion and derived shortcuts are compared with the independent solver, including on incorrect submitted tables.

The arbitrary-retention test enumerates all 121 two-vertex arenas in its declared move/exit family and all 58,564 retention and cell assignments with endpoints at most four. It accepts exactly 3,036 correct assignments and rejects all 55,528 incorrect ones. Another 2,500 seeded arenas of at most 14 vertices give 661,298 predicate comparisons and 13,452 cell mutations, with no discrepancy. Exhaustive boundary choices on 180 small models verify all 1,308 scalar ranges in Theorem 3.1, not just their extreme values. A five-vertex example with an omitted cycle, one retained cell of width four, and exit toll $10^{100} + 1$ is also certified without iterating over budgets.

For checkpoints, all 5,115 grid/period pairs with caps at most ten and periods at most five are compared with 424,945 forced-path queries. Their maximum depths equal (12). The tests also check 20,631 applicable density bounds and 5,600 instances of (14). The optimizer agrees with brute force on 120 integer-cost problems, including negative costs, across 17,763 admissible grids. These finite checks support the proofs but are not a formal verification of them. Full parameters and results are included in the artifact.

7.2 Corpus and comparison

The base corpus is every pawnless three- and four-man material inventory up to colour orientation: 24 classes with 50,289,569 valid canonical positions. Two complete five-man classes test different structure. KNNNvK exercises three identical units; KBBvKN exercises opposing material and phases beyond the clock threshold. They are selected classes, not a representative sample of five- or eight-man values.

Complete five-man class	KNNNvK	KBBvKN
Valid canonical positions	30,261,917	85,412,684
Legal edges	351,744,616	1,155,240,324
Maximum phase rank	42	132
Independent Python move sets	10,000	13,000
Edges in those move comparisons	115,539	164,039

Table 1: Complete generated sources. The producer and forward rank audit share C++ move generation; the sampled Python move generator is separate.

Retrograde events generate the exact sources; a separate forward audit checks their sign and phase equations. KBBvKN has 14,647,142 mover wins, 3,875,033 cursed wins, 50,962,387 draws, 7,665,338

blessed losses, and 8,262,784 mover losses, including terminals under the declared convention. Its cursed cases arise from the present phase; the required four-man reset outcomes are strict or drawn. Downstream cursed-reset tolls are covered by the generic theory and retained abstract tests, not this chess population.

The native size reference is the published original-format WDL/DTZ inventory in python-chess for the 24 smaller classes [9] and a pinned distribution manifest for the two five-man classes [10]. The comparison does not regenerate native tables with ply-accurate settings, compare every native distance numerically, or time a native prober. It compares complete file sizes for the same material domains and the declared query semantics, not binary compatibility or a controlled comparison against a native rebuild with identical distance precision.

7.3 Storage and ablations

Table 2 counts headers, directories, predictors, grid tags, frames, and all required lower-class files once. No separate outcome channel is added. Executable code is excluded on both sides. The complete inventory in Appendix A includes the four classes on which the final file is larger than its native pair.

Population	Native pairs (bytes)	Compressed (bytes)	Saving
All 24 pawnless three-/four-man classes	2,668,032	1,153,211	56.78%
KNNNvK	3,907,424	1,643,247	57.95%
KBBvKN	13,201,312	4,789,174	63.72%
All 26, dependencies counted once	19,776,768	**7,585,632**	**61.64%**

Table 2: Complete data lengths. Individual five-man rows exclude repeated dependency charges; their dependencies are already present in the 24-class row.

Both stored owners and coordinate orders are screened on the smaller classes before complete encoding. Five-man layout selection uses 128 seeded blocks and a declared order menu, followed by encoding of the entire chosen layout. Reported sizes are complete files, not estimates from those blocks. Using known labels to choose an encoding is not a prediction experiment on unseen data.

Complete 24-class representation	Bytes
Signed checkpoint control, maximum spacing eight	1,741,881
Reconstruction rules and spatial layout, before predictors	1,183,861
Same representation, after predictor selection	**1,153,211**
Same layout with exact cells, before predictors	3,052,634
Same layout with spacing-four cells, before predictors	2,058,607

Table 3: Within-implementation controls. The adaptive formats retain the same maximum spacing; exact and spacing-four cells trade larger files for less refinement.

The final smaller-class data are 33.80% below the signed checkpoint control, with the same seven-ply quiet bound. The controls show that spatial coding alone does not remove the storage–refinement trade-off. Five-man ablations also contain a negative result: the ordinary compact index makes KBBvKN larger. Sector separation and erasure, not compact indexing alone, account for its later improvement (Table 4). The final one-byte cell conversion preserves all 44,111,760 completed indexed symbols across the two files.

7.4 Certification of the released files

The complete file checker reads only the deployed signed files. It reconstructs omitted entries, inverts indexes and filters, derives missing-side intervals, and checks sign consistency and both forced endpoints.

Five-man representation	KNNNvK bytes	KBBvKN bytes
Raw-coordinate spatial cells, two-byte alphabet	3,133,642	8,346,290
Compact colex cells, two-byte alphabet	2,318,207	8,441,513
Compact cells with predictors, two-byte alphabet	1,868,967	8,287,836
Bishop-colour order, no sector erasure	—	6,812,736
Sector erasure with predictors, two-byte alphabet	—	5,991,524
Final one-byte cells, explicit extended rank cap	**1,643,247**	**4,789,174**

Table 4: Complete payload ablations. The final row narrows interval symbols, not exact ranks.

Lower-class outcome caches are constructed after their signed dependencies have been certified. No source rank, reconstruction mask, or input WDL file is read.

Across the 26 classes it certifies 165,964,170 valid canonical positions, performs 73,713,956 stored-side sign checks and 89,440,342 forced endpoint tests, and visits 7,493,991,298 generated edges. Every file passes. The same-colour-sector certificate is a separate prerequisite; its scan is not included in those endpoint counts. Twelve altered files are rejected, including wrong signs, false mates, incorrect rank-one claims, inflated cells, invalid index fields, truncation, and trailing data.

The five-man checker uses a resident raw-index memo, not the compact probe index. Its largest predicate allocation is 423,886,848 bytes; derived metadata takes 1,211,105,280 bytes and certified lower outcomes at most 11,590,656 bytes. Compressed input, decoded frames, and process overhead are additional. The compact archive size is not a bound on checker memory.

The producer and complete checker share C++ legal-move generation. The separate Python implementation agrees on 23,000 five-man move sets and 279,578 edges. The first 10,000 contexts per class are sampled; the extra 3,000 KBBvKN contexts are the first encountered over-threshold source entries, not a uniform cursed-position sample. These checks test an independent implementation but do not establish exhaustive external tablebase agreement. Re-encoding from complete exact sources and recorded layout choices reproduces all 26 final files byte-for-byte with Zstandard 1.5.7.

7.5 Query cost

The tested query interface accepts fresh, rights-free pawnless positions in the 26 classes, with either colour orientation and clocks zero through 99. It rejects unsupported material, castling, pawns, en-passant fields, and higher clocks. The benchmark alone reads source ranks to construct requests and expected answers; the decoder receives none of those arrays.

Uniform workloads choose nonmated valid canonical roots, including stalemates, and reserves one through 100. Threshold-adjacent requests use $z - 1, z, z + 1$ where supported. For KBBvKN, budgets above 100 compare phase ranks, not physical entering clocks. Compressed files are resident, each loaded class has a 512-KiB decoded-block allowance, and query-local transpositions are cleared between roots. There is no persistent answer cache or exact-rank fallback.

All 33,015 configuration/query evaluations agree with their audited expectations. The control and final decoder use matching request fingerprints, and their run order alternates across three repetitions. Repetitions are not independent positions. No tested query reaches the two-million-node guard or exceeds the quiet-depth bound; the guard returns no result rather than a guessed value.

Smaller files do not consistently give faster queries. KQvKR is slower than the control, and the five-man threshold tails reach roughly 79 and 133 milliseconds. These resident-file measurements do not predict cold-device or eight-man latency.

Class	Control Threshold p99	Final spatial decoder Uniform p99	Final spatial decoder Threshold p99
KBBvKN	—	1.217	133.271
KBNvK	14.374	1.958	13.849
KNNNvK	—	10.474	79.450
KQvK	4.500	1.252	5.175
KQvKQ	0.436	0.245	0.430
KQvKR	20.836	1.345	66.868
KRvKB	0.833	0.127	0.570
KRvKN	3.415	0.132	2.207

Table 5: Milliseconds per value/predicate query; medians of three run-specific p99 values. The signed control receives identical small-class requests. Neither column times native Syzygy.

8 Scope beyond the measured classes

The reset structure extends to eight-man chess and beyond. Record placement, mover, castling rights γ, and an en-passant field when a legal capture is available. Captures and pawn moves reset the clock. The lexicographic potential

$$\Phi(p) = \big(\mathrm{men}(p), \mathrm{pawns}(p), \mathrm{advance}(p), |\gamma|\big)$$

weakly decreases along legal moves and strictly decreases on resets. A white pawn on rank r contributes $8 - r$ to advance, and a black pawn contributes $r - 1$. Captures lower the man count, noncapturing promotions lower the pawn count, and other pawn moves lower advance. Castling rights never increase [16].

The representation and certification theorems therefore apply to the corresponding fresh-clock arena, provided all terminal and reset boundaries are covered. A capture may preserve castling rights; a noncapturing promotion stays within the same man count. Rights-free lower tables do not supply those missing states [2]. The theoretical arena must include them, even though the implemented decoder rejects them. Arbitrary entering repetition histories also lie outside the declared fresh-clock model.

This extension says nothing about the encoded lengths of unmeasured classes. A complete archive has size $\sum_C \ell_C + A$, where ℓ_C counts each class once and A includes any additional required data. The observed ratio 0.38356 does not bound that sum at seven or eight men. The results establish exact reconstruction, certification, and a sharp checkpoint tradeoff; target-scale storage and query costs still require target-scale evidence.

A Complete storage inventory

Class	Native pair	Compressed	Saving (%)
KBBvK	194,272	63,158	67.49
KBBvKN	13,201,312	4,789,174	63.72
KBNvK	469,472	185,327	60.52
KBvK	160	129	19.37
KBvKB	1,312	4,108	-213.11
KBvKN	2,336	4,083	-74.79
KNNNvK	3,907,424	1,643,247	57.95
KNNvK	1,440	129	91.04
KNvK	160	129	19.37
KNvKN	1,248	4,085	-227.32
KQBvK	144,032	92,275	35.93
KQNvK	160,032	96,662	39.60
KQQvK	32,992	129	99.61
KQRvK	48,672	4,115	91.55
KQvK	5,600	1,140	79.64
KQvKB	201,888	83,118	58.83
KQvKN	172,704	76,085	55.94
KQvKQ	23,264	20,231	13.04
KQvKR	329,504	176,757	46.36
KRBvK	264,608	81,647	69.14
KRNvK	297,952	121,498	59.22
KRRvK	55,456	11,729	78.85
KRvK	8,480	1,890	77.71
KRvKB	42,848	25,643	40.15
KRvKN	193,248	82,121	57.50
KRvKR	16,352	17,023	-4.10
Total	19,776,768	7,585,632	61.64

All 26 inventories are retained, including every unfavorable comparison. Native lengths describe published original-format WDL/DTZ pairs. The compressed column includes every required lower-class dependency once and no separate outcome table. Executable code is excluded on both sides.

B Reproduction

The artifact contains the 26 final files, the inputs needed to repeat the final lossless encoding step, certificate records, an independent move parser, and code. Build the chess tools with C++17 and libzstd; re-encoding also uses NumPy. From the artifact root, run

```
make implementation
python implementation/reproduce_spatial.py
python implementation/reproduce_spatial.py --full
python implementation/reencode_release.py
```

The full certificate reads signed files only; the sector prerequisite reads lower-class files only. Re-encoding starts from the retained unpredicted frames and compares every final byte. The exact five-man witnesses and the intermediate files needed to repeat the original retrograde calculation and layout search are not distributed. The supplied verification records document the earlier complete-source re-encoding. Query timings are retained measurements and are not rerun by these commands.

The general reference tools require only Python 3.10 or later. From the artifact root, run

```
python theory/verify.py --output verification/theory_results.json
python theory/verify_partial.py --output verification/partial_results.json
```

The drivers use seeds 20260906 and 20260907, respectively. Their exact solvers are separate from the certificate implementations but share arena data structures. Integer-cost tests use exact arithmetic; entropy examples use binary64 and do not predict compressed lengths.

Move generation, coverage, index and filter inverses, and the checker remain executable assumptions. Reconstruction labels need not be trusted as correct values: the certificate checks them. Hashes establish file identity, not game-theoretic correctness.

References

[1] R. de Man. Syzygy endgame tablebase generator. Repository documentation, including single-sided DTZ and the `-p` option. `https://github.com/syzygy1/tb`. Accessed 6 September 2026.

[2] N. Fiekas. python-chess: Syzygy endgame tablebase probing. Documentation, version 1.11.2. `https://python-chess.readthedocs.io/en/stable/syzygy.html`. Accessed 6 September 2026.

[3] J. A. Tamplin and G. McC. Haworth. Chess endgames: data and strategy. In *Advances in Computer Games*, pp. 81–96, 2004. doi:10.1007/978-0-387-35706-5_6.

[4] N. R. Sturtevant, A. Felner, and M. Helmert. Value compression of pattern databases. In *AAAI*, pp. 912–918, 2017. doi:10.1609/aaai.v31i1.10665.

[5] D. Gomboc, C. R. Shelton, A. S. Miner, and G. Ciardo. Comparing lossless compression methods for chess endgame data. In *ECAI*, pp. 4116–4123, 2024. doi:10.3233/FAIA240982.

[6] K. Masui and contributors. Bitshuffle: filter for improving compression of typed binary data. Repository documentation. `https://github.com/kiyo-masui/bitshuffle`. Accessed 6 September 2026.

[7] L. Ibarria, P. Lindstrom, J. Rossignac, and A. Szymczak. Out-of-core compression and decompression of large *n*-dimensional scalar fields. *Computer Graphics Forum*, 22(3):343–348, 2003. doi:10.1111/1467-8659.00681.

[8] D. P. Bertsekas. Robust shortest path planning and semicontractive dynamic programming. *Naval Research Logistics*, 66(1):15–37, 2019. doi:10.1002/nav.21697.

[9] N. Fiekas and contributors. Regular Syzygy reference files, published file-size inventory. `https://github.com/niklasf/python-chess/tree/master/data/syzygy/regular`. Per-file provenance recorded 6 September 2026 in the accompanying artifact.

[10] Gentoo contributors. Syzygy distribution manifest, pinned revision. File lengths and digests for `KNNNvK` and `KBBvKN`. `https://github.com/gentoo/gentoo/blob/f4e014b0a59fc646c647233c31f00fcd1fb907d6/games-board/tablebase-syzygy/Manifest`. Retrieved 6 September 2026. Metadata only; native payloads were not probed.

[11] Y. Björnsson, J. Schaeffer, and N. R. Sturtevant. Partial information endgame databases. In *Advances in Computer Games*, LNCS 4250, pp. 11–22, 2006. doi:10.1007/11922155_2.

[12] J. Hurd. Formal verification of chess endgame databases. In J. Hurd, E. Smith, and A. Darbari, editors, *Theorem Proving in Higher Order Logics: Emerging Trends*, pp. 85–100. Technical Report PRG-RR-05-02, Oxford University Computing Laboratory, 2005. `https://www.gilith.com/papers/chess.pdf`.

[13] A. Pavlov. Capture-quiet decomposition: a verification theorem for chess endgame tablebases. Preprint, arXiv:2604.07907v1, 9 April 2026. doi:10.48550/arXiv.2604.07907.

[14] D. E. Knuth. A generalization of Dijkstra's algorithm. *Information Processing Letters*, 6(1):1–5, 1977. doi:10.1016/0020-0190(77)90002-3.

[15] M. Bardi and J. P. Maldonado López. A Dijkstra-type algorithm for dynamic games. *Dynamic Games and Applications*, 6:263–276, 2016. doi:10.1007/s13235-015-0156-0.

[16] FIDE. Laws of Chess, effective 1 January 2023. Articles 3.7–3.8. `https://handbook.fide.com/chapter/E012023`. Accessed 6 September 2026.