Certified compression of endgame tables

MANTIS DISCOVERY

MANTIS · MIT CSAIL

OCTOBER 2026

01

The question

Endgame tables store the exact value of every position, and they are large. Can we keep only part of a table, still answer every exact query from it, and certify that the answers are right?

02

How we are approaching it

- Store signed distance intervals on part of the game graph.
- Recover the rest from two extremal boundary problems, without widening any interval.
- Certify every stored interval at both of its endpoints.
- Place checkpoints in a periodic pattern, which bounds how deep a lookup has to search.

03

King and rook against king

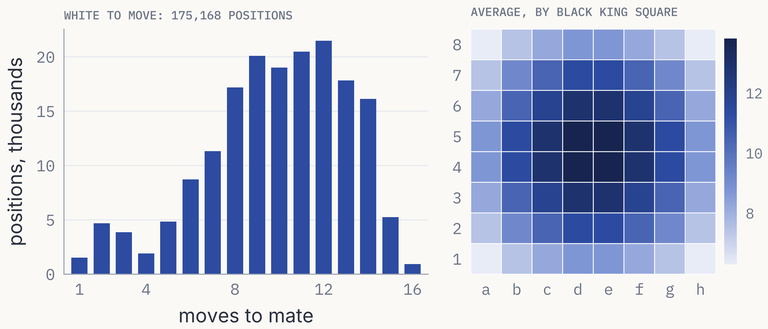


FIGURE 1. We solved this small pawnless class by retrograde analysis for the poster. The left panel counts the positions with White to move that are mated in each number of moves, and the longest mate takes 16. The right panel averages over the square of Black's king: mates take longest from the center.

04

Storage

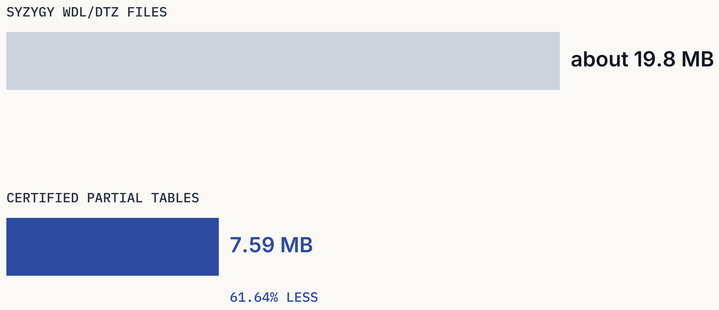


FIGURE 2. Our implementation stores 26 complete pawnless chess classes in 7.59 MB, compared with about 19.8 MB for the published Syzygy WDL/DTZ files of the same classes. We derived the Syzygy figure from the 61.64% reduction.

05

What we have so far

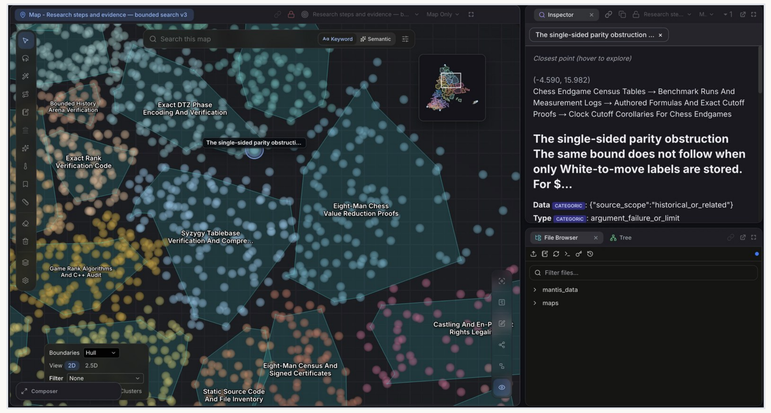
We have a sound and complete certificate for any stored subset of a finite alternating game with irreversible budget resets. It works even when the omitted part contains cycles.

Certification takes $O\big((w+\log(N+2))(N+E)\big)$ graph operations for intervals of width at most w, and $O(w(N+E))$ when the omitted part is acyclic.

The 26 classes fit in 7,585,632 bytes with all five outcome categories and exact values kept, and every file passes certification.

06

In Mantis



9,047 RECORDS · THE OPEN RECORD

The single-sided parity obstruction.

When only one side's labels are stored, a forced 99-ply path makes the grid search all 99 plies.

ALSO ON RECORD

An explicit certificate verifies a win at clock 98 but not at 99.

07

Open questions

- Near the threshold, lookups still take 133 ms at the 99th percentile, despite the seven-ply depth bound. How can that come down?
- The savings come from the chess implementation; the general checker and grid optimizer were tested separately. What does the general method save on other games?