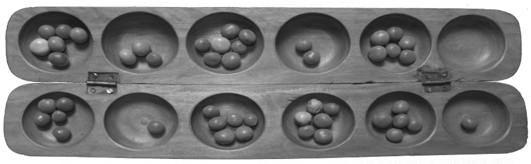
Solving the Oware, provably

Pierre-Évariste Dagand
Whisper team – Cnrs / Inria Paris

Abstract

This internship offers to implement a provably-correct solver for the game of Oware. It will take place in the Whisper team of Inria Paris – LIP6, located at University Paris 6, and will be supervised by Pierre-Évariste Dagand (Cnrs).

Last year's admission exam for École Normale Supérieure involved a challenging algorithmic task: solving a trimmed down version of the game of Oware. Quoting the subject, "the Oware is a combinatorial game with perfect information whose origin is generally associated with the Ashanti federation, in West Africa. [...] The Oware is traditionally played on a board composed of 6 pits for the player and 6 pits for her opponent."



While solving the simplified version was arduous, tackling the original Oware is challenging [1, 2, 3, 4]. First, it involves an intricate algorithm, whose correctness argument is not trivial. Second, it produces a large amount of data, which is unlikely to fit in main memory and must therefore be compactly yet efficiently stored on disk. Finally, the computation itself takes a (very) long time to complete, which makes for a slow "edit - compile - test" cycles.

Objective: Rather than burning away hundreds of CPU hours to extensively test a solver, we would like to implement a correct-by-construction solver, thus trading CPU hours for students hours. Concretely, this project entails:

- implementing a naïve, reference implementation of Oware solver;
- identifying the bottlenecks and addressing them;
- providing a formal specification and, time permitting, a correctness proof of the naïve implementation.

Overall, we are aiming for an efficient implementation (taking advantage of bit-level tricks and micro-architectural knowledge) whose correctness is justified by a mechanized proof. In particular, we wish to illustrate the fact that formal proofs provide a safety net over which we can confidently develop "clever" code without suffering from the pain of testing complex corner cases: in a context where debugging is extremely time-consuming, we want to write bug-free code!

Student's profile: We are looking for a student interested in algorithmics, program optimization and formal proofs. The solver will be implemented in OCaml, Rust or Coq. Acquaintance with an interactive theorem prover (Coq, or Isabelle) is welcome. Nonetheless, a motivated student with a strong background in functional programming (OCaml, or Haskell) could certainly learn to use Coq along the way [5, 6]. This work is funded by the Émergence(s) program of the City of Paris, thanks to which we can offer a stipend ("gratification") to an "auditeur/rice" for the duration of the internship.

References

[1] Henri Bal and Victor Allis. Parallel retrograde analysis on a distributed system. In *SC'95*, pages 73–73, 1995.

[2] Roel van der Goot. *Awari Retrograde Analysis*, pages 87–95. Springer, 2001.

[3] Thomas Robert Lincke. *Exploring the computational limits of large exhaustive search problems*. PhD thesis, Technische Wissenschaften ETH Zürich, Zürich, 2002.

[4] John Romein and Henri Bal. Solving the game of awari using parallel retrograde analysis. *IEEE computer*, 36(10):26–33, 2003.

[5] Benjamin C. Pierce, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjoberg, and Brent Yorgey. *Software Foundations*. Electronic textbook, 2014. URL `http://www.cis.upenn.edu/~bcpierce/sf`.

[6] Adam Chlipala. *Certified Programming with Dependent Types*. MIT Press, 2011. URL `http://adam.chlipala.net/cpdt/`.