

REU09: HOW HARD IS IT TO REFEREE A PARITY GAME?

SASHA RUBIN

A parity game is played on a finite directed graph for which each vertex is pre-labelled by a natural number. Two players take turns moving a token along the edges. The play runs forever (we assume no dead-ends). The first player wins the play if the smallest number occurring infinitely often in the play is even; otherwise the second player wins.

For any given game-arena (graph + vertex-labelling + starting vertex) it turns out that the game can not end in a tie. In other words, exactly one of the players has a *winning* strategy - namely, a strategy that results in a winning play no matter how the opponent moves.

Maybe surprisingly (there are infinitely many strategies!) we can decide which of the players has a winning strategy, and even construct a winning strategy for that player. This project involves looking for algorithms \mathcal{A} that, given a game-arena as input, do this fast:

Is it possible to bound the number of steps (executed by \mathcal{A})
by a polynomial in the size of the input graph?

Here the size of the graph is taken to be the number of vertices n .

For instance, there is a straightforward deterministic algorithm that requires exponentially many steps, and even one that is subexponential [1].

Although simple to state, this is a longstanding open problem in theoretical computer science. Parity games appear naturally in a number of settings: infinite duration two player games (stochastic games, mean-payoff games), verification (model-checking the modal μ -calculus), formal languages (complementing tree-automata and Rabin's famous decision procedure for S2S). These areas suggest related problems that we can look at.

Requirements: A solid background in discrete mathematics including graph theory. Preferable but not essential: programming skills and a course in design & analysis of algorithms.

References

- [1] Jurdziński and Paterson and Zwick *A Deterministic Subexponential Algorithm for Solving Parity Games* SIAM Journal on Computing, 2008.