

Solving parity games

Definition (Parity game)

$$\mathcal{G} = \langle V_E, V_A, R, \Omega : V \rightarrow \mathbb{N} \rangle$$

where

- V_E is the set of vertices for Eve,
- V_A is the set of vertices for Adam,
- $R \subseteq V \times V$ is the move relation,
- Ω defines the parity winning condition.

Definition (Winning region)

A **winning region** for Eve is the set of vertices from which she has a winning strategy.

Problem

Given a parity game, find winning regions for Eve and for Adam.

Some definitions

Definition (Attractor, Trap)

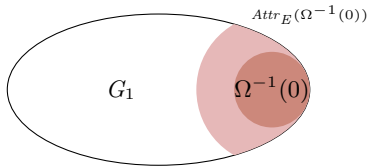
Given a set S of nodes, let $Attr^E(S)$ be the set of nodes from which Eve has a strategy to reach S .

A set S **traps Eve** if Adam has a strategy to keep Eve in S .

Remarks

- The winning region of Eve traps Adam.
- $G - Attr_E(S)$ traps Eve.

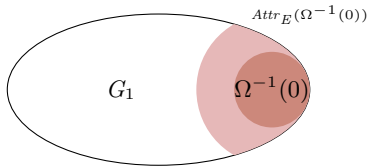
Basic algorithm



Algorithm

- Solve G_1 . Let (W_E^1, W_A^1) be the result.
- If $W_A^1 = \emptyset$ then we are done.
- If $W_A^1 \neq \emptyset$ then solve G_2 . Let (W_E^2, W_A^2)
- Return $(W_E^2, Attr_A(W_A^1) \cup W_A^2)$.

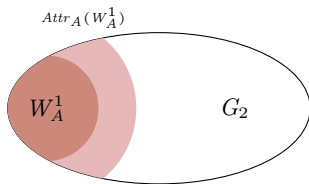
Basic algorithm



Algorithm

- Solve G_1 . Let (W_E^1, W_A^1) be the result.
- If $W_A^1 = \emptyset$ then we are done.
- If $W_A^1 \neq \emptyset$ then solve G_2 . Let (W_E^2, W_A^2)
- Return $(W_E^2, Attr_A(W_A^1) \cup W_A^2)$.

Basic algorithm



Algorithm

- Solve G_1 . Let (W_E^1, W_A^1) be the result.
- If $W_A^1 = \emptyset$ then we are done.
- If $W_A^1 \neq \emptyset$ then solve G_2 . Let (W_E^2, W_A^2)
- Return $(W_E^2, Attr_A(W_A^1) \cup W_A^2)$.

Basic algorithm

Algorithm

- Solve G_1 . Let (W_E^1, W_A^1) be the result.
- If $W_A^1 = \emptyset$ then we are done.
- If $W_A^1 \neq \emptyset$ then solve G_2 . Let (W_E^2, W_A^2)
- Return $(W_E^2, Attr_A(W_A^1) \cup W_A^2)$.

Complexity: $\mathcal{O}(2^n)$ $T(n) = T(n-1) + T(n-1)$

Complexity: $\mathcal{O}(2^d)$ where d is the number of priorities used

Finding small winning sets

Definition (Dominion)

E-dominion is a set S of vertices such that Eve has a winning strategy from all of them without leaving S . Similarly for **A-dominion**.

Dominion is either A-dominion or E-dominion.

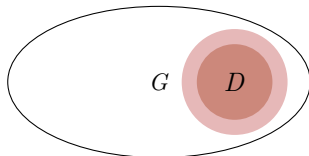
Lemma

One can find (if it exists) a dominion of size $l < n/3$ in time $\mathcal{O}(2^l \binom{n}{l})$.

Proof

- There are at most $\binom{n}{l}$ sets of size l .
- To check that a given set is a dominion takes $\mathcal{O}(2^l)$ time

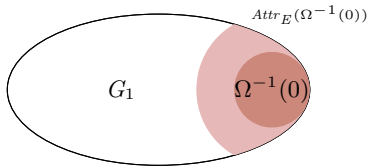
Faster algorithm



Faster algorithm [Jurdzinski, Peterson, Zwick]

- Given G find if there is a dominion of size $\leq l = \sqrt{n}$.
- If there is one, call it D , then solve $G - Attr_P(D)$.
- If not then solve $G - Attr_E(\Omega^{-1}(0))$. Let (W_A^1, W_E^1) be the result
- If $W_A^1 \neq \emptyset$ then $W_A^1 > \sqrt{n}$.
- Apply the algorithm recursively to $G - Attr_A(W_A^1)$.

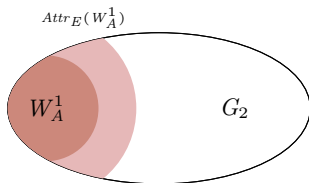
Faster algorithm



Faster algorithm [Jurdzinski, Peterson, Zwick]

- Given G find if there is a dominion of size $\leq l = \sqrt{n}$.
- If there is one, call it D , then solve $G - Attr_P(D)$.
- If not then solve $G - Attr_E(\Omega^{-1}(0))$. Let (W_A^1, W_E^1) be the result
- If $W_A^1 \neq \emptyset$ then $W_A^1 > \sqrt{n}$.
- Apply the algorithm recursively to $G - Attr_A(W_A^1)$.

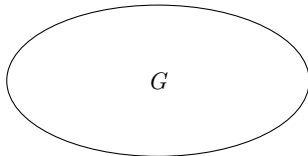
Faster algorithm



Faster algorithm [Jurdzinski, Peterson, Zwick]

- Given G find if there is a dominion of size $\leq l = \sqrt{n}$.
- If there is one, call it D , then solve $G - Attr_P(D)$.
- If not then solve $G - Attr_E(\Omega^{-1}(0))$. Let (W_A^1, W_E^1) be the result
- If $W_A^1 \neq \emptyset$ then $W_A^1 > \sqrt{n}$.
- Apply the algorithm recursively to $G - Attr_A(W_A^1)$.

Faster algorithm



Faster algorithm [Jurdzinski, Peterson, Zwick]

- Given G find if there is a dominion of size $\leq l = \sqrt{n}$.
- If there is one, call it D , then solve $G - Attr_P(D)$.
- If not then solve $G - Attr_E(\Omega^{-1}(0))$. Let (W_A^1, W_E^1) be the result
- If $W_A^1 \neq \emptyset$ then $W_A^1 > \sqrt{n}$.
- Apply the algorithm recursively to $G - Attr_A(W_A^1)$.

Complexity: $n^{\sqrt{n}}$