Perspective Games

Now you see me, now you don’t

Shubh Sharma, Siddhant Agarwal

2025-11-25

Chennai Mathematical Institute

Introduction

Perspective Games

Partial Information games : Games where players don't have all the information about the game arena and game state at all times.

Transverse Uncertainty : Sometimes player see all the information, some times they do not, like:

- Concurrent threads given control by a scheduler
- Clients connecting to a server, one at a time.
- Games with concurrent moves

Perspective Strategies or P-strategies are strategies, depending only on the part of the run that is within a player's domain, given by $\rho : V_1^{\oplus} \to V_1$.

Formal Description

Game Graph $G = \langle V_1, V_2, E, v_0, \mathrm{AP}, \tau \rangle$

- AP is a finite set of atomic propositions.
- $\tau : V_1 \sqcup V_2 \to 2^{\mathrm{AP}}$ assigns which atomic propositions are true in which locations.

Given a run $\rho = (\rho_1 \cdot \rho_2 \cdot \rho_3 \cdots) : V^\omega$, the computation of a run $\tau(\rho) : \left(2^{\mathrm{AP}}\right)^\omega$ is defined as

$$\tau(\rho) = \tau(\rho_1) \cdot \tau(\rho_2) \cdot \tau(\rho_3) \cdots$$

Winning condition L decides which runs are winning, given as one of the following:

- An LTL formula
- An ω-automata

Thus $L \subseteq \left(2^{\mathrm{AP}}\right)^\omega$.

Perspective Strategies

Given a game graph G, let the set of all finite runs on it be defined as $V^{\circledast}$.

Given a finite run ρ, we define $\pi_p : V^{\circledast} \to V_p^*$ as the projection map, which drops all vertices in the run not belonging to player p.

> A Perspective strategy σ_p for a player p takes the part of a run restricted to player p vertices and decides a move based on that.
>
> Thus, given a function $f : V_p^* \to V_p$, the following describes a P-strategy:
>
> $$\sigma_p(\rho) :\equiv f(\ \pi_1(\rho)\ )$$

We consider games where each player can have a different kind of strategy,eg. in a PF-game, player 1 has a perspective strategy and player 2 has an ordinary (or full) strategy.

Deterministic Setting

We say that a strategy σ is winning for player 1, if for any strategy τ for player 2, the play induced by the strategies satisfies the winning condition.

Some nice theorems:

- Given a game $\mathcal{G}$
  - Player 1 FF-wins $\mathcal{G}$ iff Player 1 FP-wins $\mathcal{G}$
  - Player 1 PF-wins $\mathcal{G}$ iff Player 1 PP-wins $\mathcal{G}$
- There is a game $\mathcal{G}$ such that player 1 F-wins $\mathcal{G}$ but does not P-win $\mathcal{G}$.
- Perspective Games are not determined.

Probabalistic Setting

A probabalistic strategy for player i is a function $V^{\circledast}V_i \to \mathcal{D}(V)$, where $\mathcal{D}(V)$ is the set of probability distributions on V. Given strategies g_1 and g_2 for both players we define

$$\mathcal{P}_{g_1,g_2}(L) = [\![\rho \text{ is in } L \mid \rho \text{ is generated using } g_1, g_2]\!]$$

A strategy σ is almost winning for player 1 if for every strategy τ we have $\mathcal{P}_{g_1,g_2}(L) = 1$.

Some nice Theorems:

- There is a game $\mathcal{G}$ that is PF-almost winning for player 1, but not P-winning.
- There is a game $\mathcal{G}$ that is PP-almost winning but not PF-almost winning for player 1.
- Perspective games are not almost-determined.

Deterministic Setting Analysis

Results

Deciding whether Player 1 P-wins and finding a P-strategy in a perspective game $\langle G, \mathcal{U} \rangle$ is $\mathrm{EXPTIME}$-complete when $\mathcal{U}$ is a universal automata with a parity or a reachability winning condition. The problem can be solved in time polynomial in $|G|$ and exponential in $|\mathcal{U}|$.

Deciding whether Player 1 P-wins and finding a P-strategy in a perspective game $\langle G, \psi \rangle$ is $2\mathrm{EXPTIME}$-complete when ψ is an LTL specification. The problem can be solved in time polynomial in $|G|$ and doubly exponential in $|\psi|$.

Outline for Upper Bound

- Perform a polynomial conversion from winning condition $\mathcal{U}$ to an alternating tree automaton $\mathcal{A}_{\mathcal{G}}$.
- Show that finding a P-strategy for Player 1 is equivalent to finding an accepting run in $\mathcal{A}_{\mathcal{G}}$.
- Solving the emptiness problem for $\mathcal{A}_{\mathcal{G}}$ gives an upper bound for finding Player 1 P-strategy.

Tree Automata

A Tree Automata is given by the following tuple

$$\mathcal{A} = \langle \Sigma, Q, q_{\text{in}}, \delta, \alpha \rangle$$

where

- $\alpha \subseteq Q^\omega$ is the winning condition.
- $\delta : Q \times \Sigma \to Q^*$ gives the list of locations corresponding to children of the tree.
- Σ is set of labels for the nodes of the tree.

A run of the automata on a tree T can be through as wapping the tree along the edges of the automata. It can be given as a function $r : T \to Q$ such that.

- $r(\text{root}_T) = q_{\text{in}}$
- If v is a node labelled by a with children $v_1, v_2 ... v_n$ such that $r(v) = q$. Let $\delta(q, a) = q_1, q_2 ... q_n$ we have $r(v_i) = q_i$.

Add Diagram

Non-deterministic and Universal tree automata transitions return a list of words. Both of these can be combined and represented uniquely by descriving an alternating tree automata where the transitions have the type:

$$\delta : Q \times \Sigma \to \mathcal{B}(Q \times D)$$

Where $\mathcal{B}(Q \times D)$ are positive boolean formulas. Note that runs of ATAs can be over more than 1 trees. One would pick a set nodes, and directions for edge labels.

Upper Bound (Parity Automata)

Consider a universal parity automata `cal` U

Lower Bound

LTL Winning Condition

Perspective ATL^* Model Checking

Perspective ATL^*

ATL^* is an extension of the logic CTL^* which captures the existence of strategies in a game. Perspective-ATL^* lets us quantify over perspective startegies.

There are 2 types of formulas

- State formulas φ

$$\varphi \ ::= \ \mathrm{AP} \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle\!\langle S \rangle\!\rangle \psi \mid \langle\!\langle S \rangle\!\rangle_p \psi$$

- And there are path formulas ψ

$$\psi \ ::= \ \varphi \mid \neg\psi \mid \psi \vee \psi \mid \bigcirc \psi \mid \psi \mathcal{U} \psi$$

There is also the logic ATL (similarly perspective-ATL), which simplifies ATL^* by forcing all path operators to be preceeded by path quantifiers. eg $\langle\!\langle 1 \rangle\!\rangle \bigcirc \bigcirc p$ is not allowed.

Model Checking

Model Checking is the problem of verifying if a given model M satisfies a given formula φ.

> The model checking problem for perspective-ATL^* is $\text{2-EXPTIME-complete}$. The model checking problem for perspective-ATL is PTIME-complete.

Conclusion

Probabalistic Setting

Structural Winning Condition

Memoryless Strategies

Thank You!

Thank you for attending our presentation. If you are interested in reading the paper, it can be found here:

https://dl.acm.org/doi/10.1145/3627705