

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 $\rho : V_1^\oplus \rightarrow V_1$.

Formal Description

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

- AP is a finite set of atomic propositions.
- $\tau : V_1 \sqcup V_2 \rightarrow 2^{\text{AP}}$ 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 $\tau(\rho) : (2^{\text{AP}})^\omega$ 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 $L \subseteq (2^{\text{AP}})^\omega$.

Perspective Strategies

Given a game graph G , let the set of all finite runs on it be defined as V^* .

Given a finite run ρ , we define $\pi_p : V^* \rightarrow V_p^*$ 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 $f : V_p^* \rightarrow V_p$, the following describes a **P-strategy**:

$$\sigma_p(\rho) \equiv f(\pi_p(\rho))$$

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 $V^{\otimes} V_i \rightarrow \mathcal{D}(V)$, where $\mathcal{D}(V)$ is the set of probability distributions on V . Given strategies g_1 and g_2 for both players we define

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

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

Alternating Tree Automata

Upper Bound

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 strategies.

There are 2 types of formulas

- State formulas φ

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

- 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 preceded 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 2-EXPTIME-complete. The model checking problem for perspective-ATL is PTIME-complete.

Other Fun Stuff

Probabalistic Setting

Structural Winning Condition

Memoryless Strategies

Thank You!
