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 Analysis

Introduction

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.

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:

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

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

Probabilistic Setting

Introduction

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:

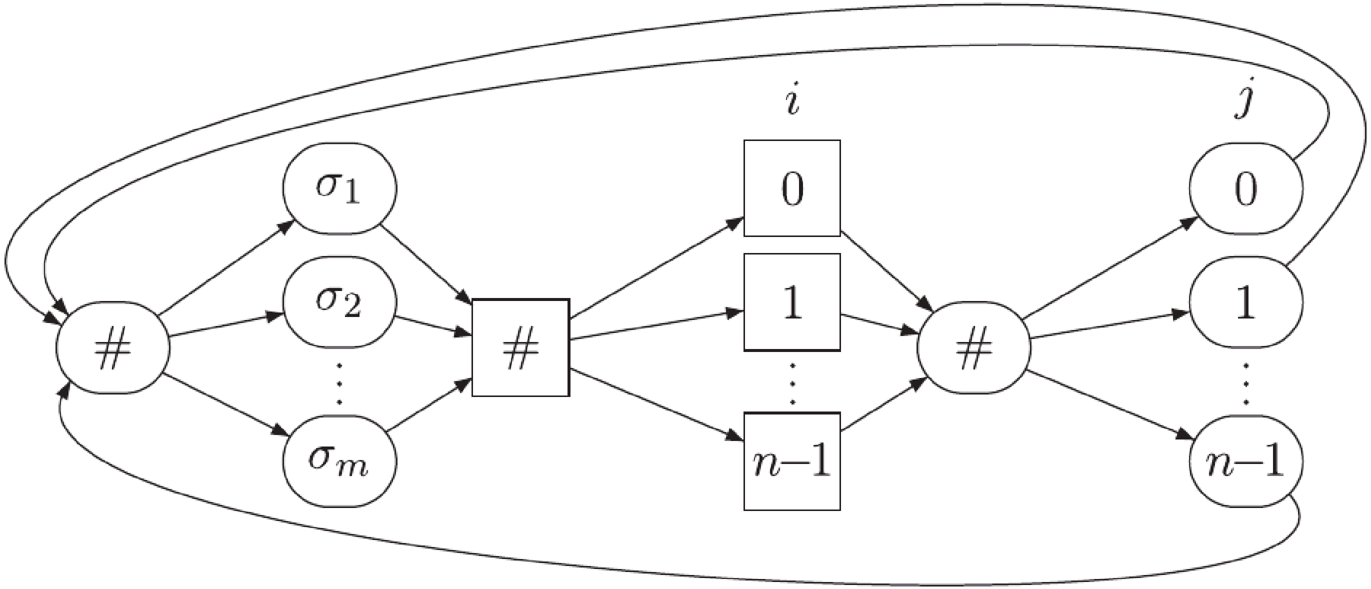
- There is a game $\mathcal{G}$ that is PF-almost winning for player 1, but not P-winning.
- There is a game $\mathcal{G}$ that is PP-almost winning but not PF-almost winning for player 1.
- Perspective games are not almost-determined.

Undecidability

Deciding whether Player 1 (P, F)-almost wins and whether they (P, P)-almost win a perspective game with a deterministic co-Büchi condition is undecidable.

- This can be shown by reduction from emptiness problem of a simplified probabilistic co-Büchi automata (PCW) which has already been proved to be undecidable.
- A simplified PCW is $\mathcal{P} = \langle \Sigma, Q, q_0, \delta, \alpha \rangle$ with the transition function $\delta : Q \times \Sigma \to \langle q_1, ..., q_n \rangle$, where $\delta(q, \sigma)$ is the tuple of states that the automata can transition to, each with $\frac{1}{n}$ probability. A word $w \in \Sigma^w$ is accepted by $\mathcal{P}$ if the acceptance probability of w in $\mathcal{P}$ (denoted $\mathrm{Prp}(w)$) is 1.
- We construct a game $\mathcal{G} = \langle G, \mathcal{A} \rangle$, where $\mathcal{A}$ is a deterministic co-Büchi automata such that $\mathcal{P}$ is nonempty iff Player 1 (P, F)-almost wins $\mathcal{G}$. Intuitively, the probabilistic transitions of $\mathcal{P}$ are simulated by randomized strategies of players in $\mathcal{G}$.

Undecidability



Undecidability

- We create $\mathcal{G}$ with a game graph that looks like the above. A play in G is an infinite sequence of rounds, such that in each round Player 1 chooses $\sigma \in \Sigma$, Player 2 chooses an index $i \in \{0, ...n-1\}$ and then Player 1 chooses an index $j \in \{0...n-1\}$
- Since Player 1 only has perspective visibility, the chose of i and j are independent. So, each player in $\mathcal{G}$ has the possibility to ensure exact simulation of the probabilistic transitions of $\mathcal{P}$ by choosing transitions to the $\{0, ..., n-1\}$ vertices in G uniformly at random. If Player 2 chooses i uniformly at random and then Player 1 chooses j without knowing i, the index $(i+j) \bmod n$ is distributed uniformly in $\{0, ..., n-1\}$.

Undecidability

- If $\mathcal{P}$ is nonempty and $w \in L(\mathcal{P})$, let g_1 be a randomized P-strategy of Player 1. Since for every random choice of Player 2 the index $(i+j) \bmod n$ is distributed uniformly, we have for every randomized strategy g_2 of Player 2, $\mathrm{Pr}_{g_1,g_2}(L(\mathcal{A})) = \mathrm{Prp}(w) = 1$
- Assume $\mathcal{P}$ is empty. Let g_2 be a randomized P-strategy of Player 2 such that $i \in \{0, ..., n-1\}$ is chosen uniformly at random. For every randomized strategy g_1 of Player 1, we have that $\mathrm{Pr}_{g_1,g_2}(L(\mathcal{A}))$ is the probability that P accepts a word w that is drawn according to some distribution that is induced g_1. Since $\mathrm{Prp}(w) < 1$ for every w, we also have $\mathrm{Pr}_{g_1,g_2}(L(\mathcal{A})) < 1$
- Every strategy discussed for Player 2 has also been perspective, so we also have $\mathcal{P}$ is nonempty iff Player 1 (P, P)-almost wins.

Conclusion

Memoryless Strategies

- Let $\mathcal{G} = \langle G, L \rangle$ be a game. Deciding whether Player 1 has a winning memoryless strategy and finding such a strategy in G is PSPACE complete when L is given by an LTL formula and is NP-complete when L is given by a reachability condition or a universal parity automata.
- When L is an LTL formula ψ, we can fix a memoryless strategy σ for Player 1 and check whether the induced graph from σ satisfies ψ or not. LTL model checking is known to be in PSPACE. The case where all vertices of G belong to Player 2 is exactly the LTL model checking problem, thus showing the hardness of finding memoryless strategies.
- To show that $< \mathcal{G}, \mathcal{A} >$ is in NP, we can do something similar and guess memoryless strategies for Player 1. We can check a guessed strategy by checking for the non emptiness of the intersection of the induced subgraph with the complement of $\mathcal{A}$. The hardness of the problem is obtained via reduction from 2DP.

Perspective ATL^* Model Checking

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

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

- And there are path formulas ψ

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

Perspective ATL^* Model Checking

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.

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

Structural Winning Condition

- The games which admit memoryless strategies (Büchi, Parity, etc) admit P-strategies as well because memoryless strategies are also P-strategies.
- Generalised Büchi games admit P-strategies. Let $G = \langle V_1, V_2, v_0, E, \alpha \rangle$ be a game with a generalised Büchi winning condition. We denote by G^v the game graph obtained by changing the initial vertex to v. Let $\alpha = \{\alpha_1, ..., \alpha_k\}$
- Let f_1 be a winning F-strategy for Player 1 in $\mathcal{G}$ and let $U \subseteq V$ be the set of vertices that are reachable when Player 1 plays according to f_1. f_1 will also be winning in $\langle G^v, \alpha \rangle$ for every $v \in U$.
- For every $1 \leq i \leq k$ and $v \in U$, f_1 induces a winning strategy for Player 1 in the Büchi game on G^v with objective $\alpha_i \cap U$

Structural Winning Condition

- Use the following P-strategy for Player 1: Start with $i = 1$, play according to the memoryless strategy of $< G^v, \alpha_i \cap U >$ while maintaining a counter which is increased every time the play visits V_1. When the counter is at least n and the play reaches a vertex v in $U \cap V_1$, Player 1 resets the counter and increases i to $i + 1 \mod k$ and starts playing the memoryless strategy of the new Büchi game $\langle G^v, \alpha_i \cap U \rangle$
- In n steps the play is guaranteed to reach a vertex in α_i, so before incrementing i, we always visit α_i. This way each set in α is visited infinitely often.

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