Two-player Reachability Games with Partial Observation on Both Sides

Hugo Gimbert, CNRS, LaBRI, Bordeaux

January 30, 2009

Motivations

- Controller synthesis.

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.
- $\Longrightarrow$ two-player game

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.
- $\Longrightarrow$ two-player game
- Partial observation for both the controller and the user.

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.
- $\implies$ two-player game
- Partial observation for both the controller and the user.
- Avoids implementation of full monitoring.

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.
- $\Longrightarrow$ two-player game
- Partial observation for both the controller and the user.
- Avoids implementation of full monitoring.
- First step towards distributed synthesis.

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.
- $\implies$ two-player game
- Partial observation for both the controller and the user.
- Avoids implementation of full monitoring.
- First step towards distributed synthesis.
- Reachability, safety (+Büchi) conditions.

Motivations

- Controller synthesis.
- System interacting with both the controller and the user.
- $\implies$ two-player game
- Partial observation for both the controller and the user.
- Avoids implementation of full monitoring.
- First step towards distributed synthesis.
- Reachability, safety (+Büchi) conditions.
- Determinacy, decidability. With Nathalie Bertrand and Blaise Genest, Rennes.

- Two-players games with reachability condition.

- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.

- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.

- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.
- They interact synchronously.

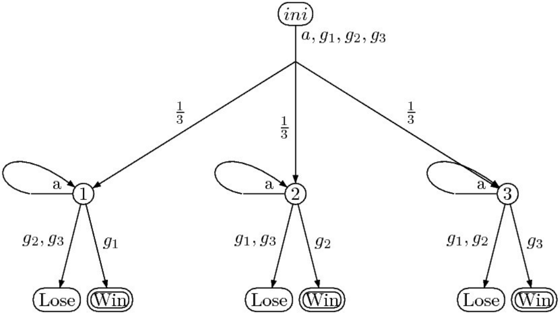
- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.
- They interact synchronously.
- Initial state k_0. Players 1 and 2 choose simultaneously actions i_{n+1} and j_{n+1}.

- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.
- They interact synchronously.
- Initial state k_0. Players 1 and 2 choose simultaneously actions i_{n+1} and j_{n+1}.
- This determines the new state k_{n+1} depending on the current state k_n and the actions i_{n+1} and j_{n+1}.

- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.
- They interact synchronously.
- Initial state k_0. Players 1 and 2 choose simultaneously actions i_{n+1} and j_{n+1}.
- This determines the new state k_{n+1} depending on the current state k_n and the actions i_{n+1} and j_{n+1}.
- Transitions may be probabilistic.

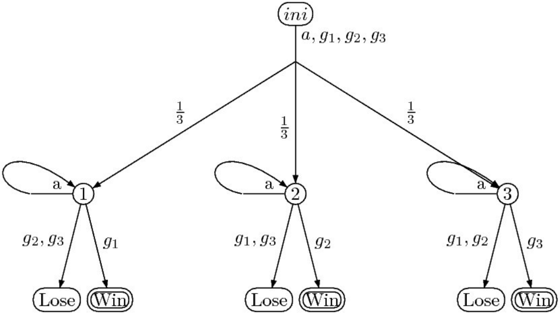
- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.
- They interact synchronously.
- Initial state k_0. Players 1 and 2 choose simultaneously actions i_{n+1} and j_{n+1}.
- This determines the new state k_{n+1} depending on the current state k_n and the actions i_{n+1} and j_{n+1}.
- Transitions may be probabilistic.
- Goal of player 1: reaching a set of target states $T \subseteq K$.

- Two-players games with reachability condition.
- Discrete-time systems. System goes through a sequence of states $k_0, k_1, \ldots \in K^*$.
- Two decision makers = player 1 and player 2.
- They interact synchronously.
- Initial state k_0. Players 1 and 2 choose simultaneously actions i_{n+1} and j_{n+1}.
- This determines the new state k_{n+1} depending on the current state k_n and the actions i_{n+1} and j_{n+1}.
- Transitions may be probabilistic.
- Goal of player 1: reaching a set of target states $T \subseteq K$.
- Goal of player 2: the opposite.



Only one player.

- States $K = \{\text{ini},1,2,3,\text{Lose},\text{Win}\}$



Only one player.

- States $K = \{\text{ini,1,2,3,Lose,Win}\}$
- Actions $I = \{a, g_1, g_2, g_3\}$

Partial observation via signals

- Set of states K, set of actions I and J for player 1 and 2.

Partial observation via signals

- Set of states K, set of actions I and J for player 1 and 2.
- Set of signals C and D for player 1 and 2.

Partial observation via signals

- Set of states K, set of actions I and J for player 1 and 2.
- Set of signals C and D for player 1 and 2.
- Current state k.

Partial observation via signals

- Set of states K, set of actions I and J for player 1 and 2.
- Set of signals C and D for player 1 and 2.
- Current state k.
- Players choose actions i and j

Partial observation via signals

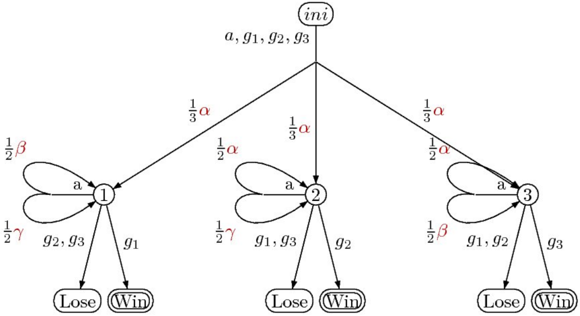
- Set of states K, set of actions I and J for player 1 and 2.
- Set of signals C and D for player 1 and 2.
- Current state k.
- Players choose actions i and j
- Players receive signals c and d.

Partial observation via signals

- Set of states K, set of actions I and J for player 1 and 2.
- Set of signals C and D for player 1 and 2.
- Current state k.
- Players choose actions i and j
- Players receive signals c and d.
- New state k' and signals c and d with probability $0 \leq p(k', c, d \mid k, i, j) \leq 1$.

Partial observation via signals

- Set of states K, set of actions I and J for player 1 and 2.
- Set of signals C and D for player 1 and 2.
- Current state k.
- Players choose actions i and j
- Players receive signals c and d.
- New state k' and signals c and d with probability $0 \leq p(k', c, d \mid k, i, j) \leq 1$.
- A play is an infinite sequence of states, actions and signals in $(KIJCD)^\omega$



Only one player.

- States $K = \{\text{ini},1,2,3,\text{Lose},\text{Win}\}$
- Actions $I = \{a, g_1, g_2, g_3\}$
- Signals $C = \{\alpha, \beta, \gamma\}$

Strategies

- After a finite play $(k_0,$

Strategies

- After a finite play $(k_0,$

Strategies

- After a finite play $(k_0, i_0, j_0,$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0,$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1,$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1,$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1,$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots,$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.
- In games with perfect information: choices depend on the whole play.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.
- In games with perfect information: choices depend on the whole play.
- In games with partial observation: choices depend only on signals.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.
- In games with perfect information: choices depend on the whole play.
- In games with partial observation: choices depend only on signals.
- Strategy σ for player 1: $\sigma_\delta : C^* \to \mathcal{D}(I)$, C signals of player 1 and $\mathcal{D}(I)$ probability distributions on actions of 1.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.
- In games with perfect information: choices depend on the whole play.
- In games with partial observation: choices depend only on signals.
- Strategy σ for player 1: $\sigma_\delta : C^* \to \mathcal{D}(I)$, C signals of player 1 and $\mathcal{D}(I)$ probability distributions on actions of 1.
- After signals $c_0 c_1 \ldots c_n$ player 1 plays action i with proba $\sigma_\delta(i)$.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.
- In games with perfect information: choices depend on the whole play.
- In games with partial observation: choices depend only on signals.
- Strategy σ for player 1: $\sigma_\delta : C^* \to \mathcal{D}(I)$, C signals of player 1 and $\mathcal{D}(I)$ probability distributions on actions of 1.
- After signals $c_0 c_1 \ldots c_n$ player 1 plays action i with proba $\sigma_\delta(i)$.
- Initial state: initial probability distribution $\delta \in \mathcal{D}(K)$.

Strategies

- After a finite play $(k_0, i_0, j_0, c_0, d_0, k_1, i_1, j_1, c_1, d_1, \ldots, k_n)$, players choose actions i_n and j_n.
- Choices are made using strategies.
- In games with perfect information: choices depend on the whole play.
- In games with partial observation: choices depend only on signals.
- Strategy σ for player 1: $\sigma_\delta : C^* \to \mathcal{D}(I)$, C signals of player 1 and $\mathcal{D}(I)$ probability distributions on actions of 1.
- After signals $c_0 c_1 \ldots c_n$ player 1 plays action i with proba $\sigma_\delta(i)$.
- Initial state: initial probability distribution $\delta \in \mathcal{D}(K)$.
- Signals contain actions.

Winning probability

- Initial distribution δ and strategies σ and τ. Markov chain. Probability measure $\mathbb{P}_{\delta}^{\sigma,\tau}(\cdot)$ on the set of infinite plays $(KIJCD)^{\omega}$.

Winning probability

- Initial distribution δ and strategies σ and τ. Markov chain. Probability measure $\mathbb{P}_{\delta}^{\sigma,\tau}(\cdot)$ on the set of infinite plays $(KIJCD)^{\omega}$.
- Player 1 wants the probability to reach a target state

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T)$$

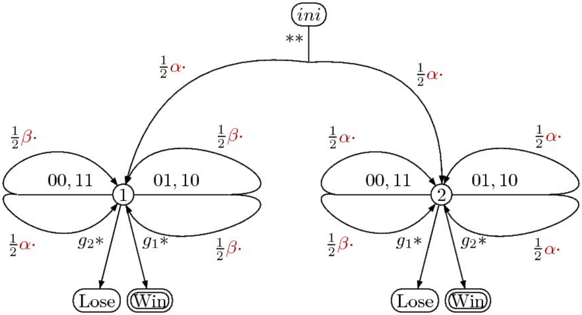
to be as close to 1 as possible.

Winning probability

- Initial distribution δ and strategies σ and τ. Markov chain. Probability measure $\mathbb{P}_{\delta}^{\sigma,\tau}(\cdot)$ on the set of infinite plays $(KIJCD)^{\omega}$.
- Player 1 wants the probability to reach a target state

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T)$$

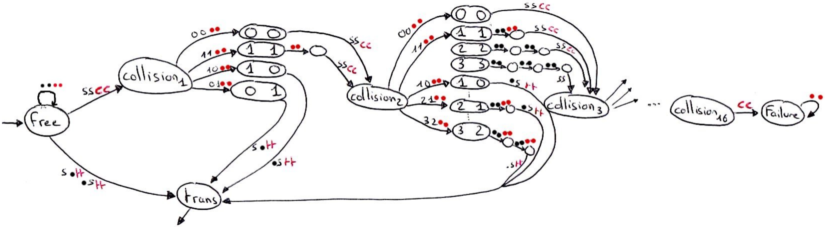
  to be as close to 1 as possible.
- Player 2 has the opposite goal.



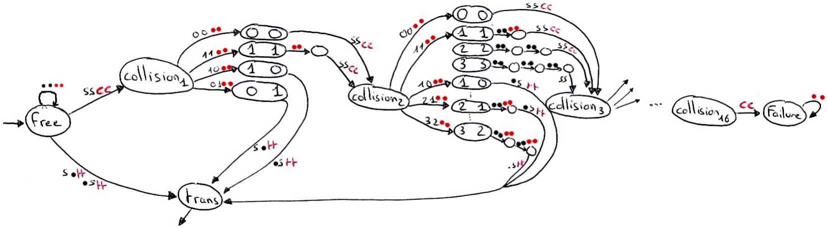
Two players.

- States K = {ini,1,2,Lose,Win}
- Actions $I = \{0, 1, g_1, g_2\}$ and $J = \{0, 1\}$
- Player 2 is blind.
- Signals $C = \{\alpha, \beta, \gamma\}$ and $D = \{\cdot\}$

- collisions occur on ethernet networks,



- collisions occur on ethernet networks,
- controllers choose to stay idle for some randomly chosen delay.



Winning almost-surely, surely and positively

Player 1 wants $\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T)$ to be as close to 1 as possible.

Winning almost-surely, surely and positively

Player 1 wants $\mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T)$ to be as close to 1 as possible.

- Player 1 wins almost-surely: if player 1 has a strategy σ such that for every strategy τ of player 2,

$$\mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) = 1 \quad .$$

Winning almost-surely, surely and positively

Player 1 wants $\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T)$ to be as close to 1 as possible.

- Player 1 wins almost-surely: if player 1 has a strategy σ such that for every strategy τ of player 2,

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 1 \ .$$

- Player 2 wins almost-surely: if $\exists \tau \; \forall \sigma \; \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 0$.

Winning almost-surely, surely and positively

Player 1 wants $\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T)$ to be as close to 1 as possible.

- Player 1 wins almost-surely: if player 1 has a strategy σ such that for every strategy τ of player 2,

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 1 \ .$$

- Player 2 wins almost-surely: if $\exists \tau \ \forall \sigma \ \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 0$.
- Player 1 wins positively:

$$\exists \sigma \ \forall \tau \ \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) > 0 \ .$$

Winning almost-surely, surely and positively

Player 1 wants $\mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T)$ to be as close to 1 as possible.

- Player 1 wins almost-surely: if player 1 has a strategy σ such that for every strategy τ of player 2,

$$\mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) = 1 \ .$$

- Player 2 wins almost-surely: if $\exists \tau \ \forall \sigma \ \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) = 0$.
- Player 1 wins positively:

$$\exists \sigma \ \forall \tau \ \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) > 0 \ .$$

- Player 2 wins positively: if $\exists \tau \ \forall \sigma \ \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) < 1$.

Winning almost-surely, surely and positively

Player 1 wants $\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T)$ to be as close to 1 as possible.

- Player 1 wins almost-surely: if player 1 has a strategy σ such that for every strategy τ of player 2,

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 1 \quad .$$

- Player 2 wins almost-surely: if $\exists \tau \ \forall \sigma \ \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 0$.
- Player 1 wins positively:

$$\exists \sigma \ \forall \tau \ \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) > 0 \quad .$$

- Player 2 wins positively: if $\exists \tau \ \forall \sigma \ \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) < 1$.
- Randomized strategies are necessary, in practice (ethernet) and in theory (determinacy).

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.
- Player 2 wins positively $(\exists\tau\forall\sigma < 1)$
  $\implies$ Player 1 does not win almost-surely $\neg(\exists\sigma\forall\tau = 1)$.

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.
- Player 2 wins positively $(\exists\tau\forall\sigma < 1)$
  $\implies$ Player 1 does not win almost-surely $\neg(\exists\sigma\forall\tau = 1)$.
- Converse?

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.
- Player 2 wins positively $(\exists\tau\forall\sigma < 1)$
  $\implies$ Player 1 does not win almost-surely $\neg(\exists\sigma\forall\tau = 1)$.
- Converse?
- $(\forall\tau\exists\sigma > 0) \implies (\exists\sigma\forall\tau > 0)$?

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.
- Player 2 wins positively $(\exists\tau\forall\sigma < 1)$
  $\implies$ Player 1 does not win almost-surely $\neg(\exists\sigma\forall\tau = 1)$.
- Converse?
- $(\forall\tau\exists\sigma > 0) \implies (\exists\sigma\forall\tau > 0)$?
- $(\forall\sigma\exists\tau < 1) \implies (\exists\tau\forall\sigma < 1)$?

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.
- Player 2 wins positively $(\exists\tau\forall\sigma < 1)$
  $\implies$ Player 1 does not win almost-surely $\neg(\exists\sigma\forall\tau = 1)$.
- Converse?
- $(\forall\tau\exists\sigma > 0) \implies (\exists\sigma\forall\tau > 0)$?
- $(\forall\sigma\exists\tau < 1) \implies (\exists\tau\forall\sigma < 1)$?
- Yes, it holds.

Qualitative Determinacy

- Player 1 wins positively $(\exists\sigma\forall\tau > 0)$
  $\implies$ Player 2 does not win almost-surely $\neg(\exists\tau\forall\sigma = 0)$.
- Player 2 wins positively $(\exists\tau\forall\sigma < 1)$
  $\implies$ Player 1 does not win almost-surely $\neg(\exists\sigma\forall\tau = 1)$.
- Converse?
- $(\forall\tau\exists\sigma > 0) \implies (\exists\sigma\forall\tau > 0)$?
- $(\forall\sigma\exists\tau < 1) \implies (\exists\tau\forall\sigma < 1)$?
- Yes, it holds.
- Requires knowledge about almost-surely and positively winning strategies.

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,
- surely winning for player 2,

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,
- surely winning for player 2,
- or positively winning for both player 1 and 2.

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,
- surely winning for player 2,
- or positively winning for both player 1 and 2.

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,
- surely winning for player 2,
- or positively winning for both player 1 and 2.

Remarks:

- Generalizes result of [deAlfaro, Henzinger 00] for full monitoring.

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,
- surely winning for player 2,
- or positively winning for both player 1 and 2.

Remarks:

- Generalizes result of [deAlfaro, Henzinger 00] for full monitoring.
- Holds for perfect-info, prefix-independent games [G., Horn 08].

Qualitative Determinacy

Determinacy Theorem: [Bertrand, Genest, G.] A partial observation reachability games with finitely many states, actions and signals is either:

- almost-surely winning for player 1,
- surely winning for player 2,
- or positively winning for both player 1 and 2.

Remarks:

- Generalizes result of [deAlfaro, Henzinger 00] for full monitoring.
- Holds for perfect-info, prefix-independent games [G., Horn 08].
- False for Büchi games.

Values

Definition: a game has a value $0 \leq v \leq 1$, if player 1 can defend probability $v + \epsilon$ and player 2 can secure probability $v - \epsilon$, for every ϵ.

$$\sup_\sigma \inf_\tau \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) = \leq v \leq \inf_\tau \sup_\sigma \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T).$$

Values

Definition: a game has a value $0 \leq v \leq 1$, if player 1 can defend probability $v + \epsilon$ and player 2 can secure probability $v - \epsilon$, for every ϵ.

$$\sup_\sigma \inf_\tau \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T) = \leq v \leq \inf_\tau \sup_\sigma \mathbb{P}_\delta^{\sigma,\tau}(\exists n, K_n \in T).$$

- Proposition [Renaut, Sorin, G.]: two players zero-sum reachability games with partial observation have a value.

Values

Definition: a game has a value $0 \leq v \leq 1$, if player 1 can defend probability $v + \epsilon$ and player 2 can secure probability $v - \epsilon$, for every ϵ.

$$\sup_{\sigma} \inf_{\tau} \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = \leq v \leq \inf_{\tau} \sup_{\sigma} \mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T).$$

- Proposition [Renaut, Sorin, G.]: two players zero-sum reachability games with partial observation have a value.
- Theorem [Paz 71, Bertoni 75]: values are not computable, even for one player and no observations.

Decidable cases

- Full monitoring: players observe everything.

Decidable cases

- Full monitoring: players observe everything.
- Values are computable (2-EXPTIME) (Chatterjee, Henzinger).

Decidable cases

- Full monitoring: players observe everything.
- Values are computable (2-EXPTIME) (Chatterjee, Henzinger).
- Existence of almost-surely or positively winning strategies decidable in polynomial time (deAlfaro, Henzinger, 00).

Decidable cases

- Full monitoring: players observe everything.
- Values are computable (2-EXPTIME) (Chatterjee, Henzinger).
- Existence of almost-surely or positively winning strategies decidable in polynomial time (deAlfaro, Henzinger, 00).
- One player perfectly informed + deterministic transitions: Existence of almost-surely or positively strategies decidable in EXPTIME (Chatterjee, Doyen, Henzinger, Raskin 03).

Positive winning for Büchi games

- Theorem [Baier, Bertrand, Groesser, 07]: Büchi games. It is undecidable whether player 1 has a positively winning strategy.

Positive winning for Büchi games

- Theorem [Baier, Bertrand, Groesser, 07]: Büchi games. It is undecidable whether player 1 has a positively winning strategy.
- Holds even for player 1 blind and alone.

Memory requirements

- Memory necessary for implementing the strategies?

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.
- Beliefs: set $L \subseteq K$ of possible current states according to signals sequence.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.
- Beliefs: set $L \subseteq K$ of possible current states according to signals sequence.
- Player 2 play an action s.t. her next belief is surely winning.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.
- Beliefs: set $L \subseteq K$ of possible current states according to signals sequence.
- Player 2 play an action s.t. her next belief is surely winning.
- Player 1 needs memory $\mathcal{P}(K)$ for winning almost-surely.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.
- Beliefs: set $L \subseteq K$ of possible current states according to signals sequence.
- Player 2 play an action s.t. her next belief is surely winning.
- Player 1 needs memory $\mathcal{P}(K)$ for winning almost-surely.
- Plays all actions s.t. his next belief is almost-surely winning.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.
- Beliefs: set $L \subseteq K$ of possible current states according to signals sequence.
- Player 2 play an action s.t. her next belief is surely winning.
- Player 1 needs memory $\mathcal{P}(K)$ for winning almost-surely.
- Plays all actions s.t. his next belief is almost-surely winning.
- Not very surprising, true for one-player games.

Memory requirements

- Memory necessary for implementing the strategies?
- Player 1 needs no memory for winning positively: play randomly all actions.
- Player 2 needs memory $\mathcal{P}(K)$ for winning surely.
- Beliefs: set $L \subseteq K$ of possible current states according to signals sequence.
- Player 2 play an action s.t. her next belief is surely winning.
- Player 1 needs memory $\mathcal{P}(K)$ for winning almost-surely.
- Plays all actions s.t. his next belief is almost-surely winning.
- Not very surprising, true for one-player games.
- Upper bounds for time and lower bounds for probabilities.

Memory requirements, the element of surprise

- What about player 2 winning positively?

Memory requirements, the element of surprise

- What about player 2 winning positively?
- She wants to ensure that with positive probability, the game will never reach a target state.

Memory requirements, the element of surprise

- What about player 2 winning positively?
- She wants to ensure that with positive probability, the game will never reach a target state.
- Player 2 needs doubly-exponential memory $\mathcal{P}(\mathcal{P}(K) \times K)$ for winning positively.

Memory requirements, the element of surprise

- What about player 2 winning positively?
- She wants to ensure that with positive probability, the game will never reach a target state.
- Player 2 needs doubly-exponential memory $\mathcal{P}(\mathcal{P}(K) \times K)$ for winning positively.
- Player 2 remembers the set of possible beliefs of player 1.

Memory requirements, the element of surprise

- What about player 2 winning positively?
- She wants to ensure that with positive probability, the game will never reach a target state.
- Player 2 needs doubly-exponential memory $\mathcal{P}(\mathcal{P}(K) \times K)$ for winning positively.
- Player 2 remembers the set of possible beliefs of player 1.
- Beliefs of 2 about beliefs of 1. When player 2 receives signal d, she makes hypothesis about signal c received by player 1.

Memory requirements, the element of surprise

- What about player 2 winning positively?
- She wants to ensure that with positive probability, the game will never reach a target state.
- Player 2 needs doubly-exponential memory $\mathcal{P}(\mathcal{P}(K) \times K)$ for winning positively.
- Player 2 remembers the set of possible beliefs of player 1.
- Beliefs of 2 about beliefs of 1. When player 2 receives signal d, she makes hypothesis about signal c received by player 1.
- Very different from perfect-information games, concurrent games, even games with a perfectly informed opponent.

Doubly exponential memory is needed

Proposition: Player 2 needs double-exponential memory for winning positively.

Special case: exponential memory if player 1 perfectly informed, because guessing current state = guessing belief of 1.

Special case: exponential memory if player 2 observes signals of player 1 because player 2 can compute the belief of player 1.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.
- Reveal E to player 1 but not to player 2.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.
- Reveal E to player 1 but not to player 2.
- Player 1 reveals to player 2 sets $E_1, \ldots, E_{\frac{1}{2}C^n_{n/2}}$ of $n/2$ objects, different from E.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.
- Reveal E to player 1 but not to player 2.
- Player 1 reveals to player 2 sets $E_1, \ldots, E_{\frac{1}{2}C^n_{n/2}}$ of $n/2$ objects, different from E.
- Player 2 has $\frac{1}{2}C^n_{n/2}$ tries for guessing correctly E.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.
- Reveal E to player 1 but not to player 2.
- Player 1 reveals to player 2 sets $E_1, \ldots, E_{\frac{1}{2} C^n_{n/2}}$ of $n/2$ objects, different from E.
- Player 2 has $\frac{1}{2} C^n_{n/2}$ tries for guessing correctly E.
- Whenever player 1 cheats or player 2 gets wrong, it gives positive probability to win to the opponent.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.
- Reveal E to player 1 but not to player 2.
- Player 1 reveals to player 2 sets $E_1, \ldots, E_{\frac{1}{2}C^n_{n/2}}$ of $n/2$ objects, different from E.
- Player 2 has $\frac{1}{2}C^n_{n/2}$ tries for guessing correctly E.
- Whenever player 1 cheats or player 2 gets wrong, it gives positive probability to win to the opponent.
- Start again.

Proposition: Player 2 needs double-exponential memory for winning positively.

Example:

- Randomly choose among n objects a set E of $n/2$ objects.
- Reveal E to player 1 but not to player 2.
- Player 1 reveals to player 2 sets $E_1, \ldots, E_{\frac{1}{2}C^n_{n/2}}$ of $n/2$ objects, different from E.
- Player 2 has $\frac{1}{2}C^n_{n/2}$ tries for guessing correctly E.
- Whenever player 1 cheats or player 2 gets wrong, it gives positive probability to win to the opponent.
- Start again.
- Implementable with polynomial number of states.

Implementing the example

Choosing randomly the set E of $n/2$ objects, revealing it to player 1 but not to player 2.

- Two counters from 1 to n and from 1 to $n/2$ are used.

Implementing the example

Choosing randomly the set E of $n/2$ objects, revealing it to player 1 but not to player 2.

- Two counters from 1 to n and from 1 to $n/2$ are used.
- They are used for generating randomly a sequence of signals in $\{0,1\}^n$ with exactly $n/2$ letters 1.

Implementing the example

Choosing randomly the set E of $n/2$ objects, revealing it to player 1 but not to player 2.

- Two counters from 1 to n and from 1 to $n/2$ are used.
- They are used for generating randomly a sequence of signals in $\{0,1\}^n$ with exactly $n/2$ letters 1.
- Moreover a counter is used for storing one of the positions m where a "1" was sent.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^{n}_{n/2}$ sets different from E.

- Player 1 is a cheater if a set F he reveals is exactly E, i.e. if every object in F is in E as well.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^{n}_{n/2}$ sets different from E.

- Player 1 is a cheater if a set F he reveals is exactly E, i.e. if every object in F is in E as well.
- For revealing set F, player 1 sends n signals in $\{0, 1\}$ to player 2, 1001 means "objects 1 and 4".

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2} C^n_{n/2}$ sets different from E.

- Player 1 is a cheater if a set F he reveals is exactly E, i.e. if every object in F is in E as well.
- For revealing set F, player 1 sends n signals in $\{0, 1\}$ to player 2, 1001 means "objects 1 and 4".
- At one of the moments where he sends a "1", player 1 announce secretly to the system that the current object m is not in the set E.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^{n}_{n/2}$ sets different from E.

- Player 1 is a cheater if a set F he reveals is exactly E, i.e. if every object in F is in E as well.
- For revealing set F, player 1 sends n signals in $\{0, 1\}$ to player 2, 1001 means "objects 1 and 4".
- At one of the moments where he sends a "1", player 1 announce secretly to the system that the current object m is not in the set E.
- In case m coincides with the position stored previously, player 1 is a cheater and he is punished by losing for sure.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2} C^{n}_{n/2}$ sets different from E.

- Player 1 is a cheater if a set F he reveals is exactly E, i.e. if every object in F is in E as well.
- For revealing set F, player 1 sends n signals in $\{0, 1\}$ to player 2, 1001 means "objects 1 and 4".
- At one of the moments where he sends a "1", player 1 announce secretly to the system that the current object m is not in the set E.
- In case m coincides with the position stored previously, player 1 is a cheater and he is punished by losing for sure.
- Player 1 may cheat without being caught, but he has positive probability to get caught.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^n_{n/2}$ sets different from E.

- Player 1 may cheat by revealing less than $\frac{1}{2}C^n_{n/2}$ sets to player 2, in that case $\frac{1}{2}C^n_{n/2}$ tries are not enough for player 2 for guessing E.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^n_{n/2}$ sets different from E.

- Player 1 may cheat by revealing less than $\frac{1}{2}C^n_{n/2}$ sets to player 2, in that case $\frac{1}{2}C^n_{n/2}$ tries are not enough for player 2 for guessing E.
- We need to count from 1 to $\frac{1}{2}C^n_{n/2}$.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^n_{n/2}$ sets different from E.

- Player 1 may cheat by revealing less than $\frac{1}{2}C^n_{n/2}$ sets to player 2, in that case $\frac{1}{2}C^n_{n/2}$ tries are not enough for player 2 for guessing E.
- We need to count from 1 to $\frac{1}{2}C^n_{n/2}$.
- A simple counter would use exponentially many states.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^n_{n/2}$ sets different from E.

- Player 1 may cheat by revealing less than $\frac{1}{2}C^n_{n/2}$ sets to player 2, in that case $\frac{1}{2}C^n_{n/2}$ tries are not enough for player 2 for guessing E.
- We need to count from 1 to $\frac{1}{2}C^n_{n/2}$.
- A simple counter would use exponentially many states.
- Instead we force player 1 to count himself, by sending to the system the value of the counter, in binary form, using actions in $\{0,1\}^*$.

Implementing the example

Forcing player 1 to reveal to player 2 $\frac{1}{2}C^n_{n/2}$ sets different from E.

- Player 1 may cheat by revealing less than $\frac{1}{2}C^n_{n/2}$ sets to player 2, in that case $\frac{1}{2}C^n_{n/2}$ tries are not enough for player 2 for guessing E.
- We need to count from 1 to $\frac{1}{2}C^n_{n/2}$.
- A simple counter would use exponentially many states.
- Instead we force player 1 to count himself, by sending to the system the value of the counter, in binary form, using actions in $\{0,1\}^*$.
- The system is able to detect with positive probability whenever player 1 is cheating with the counter.

- If player 1 cheats once, player 2 has positive probability to win.

- If player 1 cheats once, player 2 has positive probability to win.
- If player 1 never cheats, and player 2 remembers all $\frac{1}{2} C^n_{n/2}$ sets revealed by player 1, then player 2 wins surely.

- If player 1 cheats once, player 2 has positive probability to win.
- If player 1 never cheats, and player 2 remembers all $\frac{1}{2} C^n_{n/2}$ sets revealed by player 1, then player 2 wins surely.
- If player 2 does not remember all $\frac{1}{2} C^n_{n/2}$ sets revealed by player 1, with positive probability she will guess wrongly the set E.

- If player 1 cheats once, player 2 has positive probability to win.
- If player 1 never cheats, and player 2 remembers all $\frac{1}{2}C^n_{n/2}$ sets revealed by player 1, then player 2 wins surely.
- If player 2 does not remember all $\frac{1}{2}C^n_{n/2}$ sets revealed by player 1, with positive probability she will guess wrongly the set E.
- Infinitely often $\implies$ player 1 wins almost-surely.

- If player 1 cheats once, player 2 has positive probability to win.
- If player 1 never cheats, and player 2 remembers all $\frac{1}{2}C^n_{n/2}$ sets revealed by player 1, then player 2 wins surely.
- If player 2 does not remember all $\frac{1}{2}C^n_{n/2}$ sets revealed by player 1, with positive probability she will guess wrongly the set E.
- Infinitely often $\implies$ player 1 wins almost-surely.
- $\implies$ Player 2 wins positively, for that she needs to remember $\frac{1}{2}C^n_{n/2}$ sets among $C^n_{n/2}$.

- If player 1 cheats once, player 2 has positive probability to win.
- If player 1 never cheats, and player 2 remembers all $\frac{1}{2}C^n_{n/2}$ sets revealed by player 1, then player 2 wins surely.
- If player 2 does not remember all $\frac{1}{2}C^n_{n/2}$ sets revealed by player 1, with positive probability she will guess wrongly the set E.
- Infinitely often $\implies$ player 1 wins almost-surely.
- $\implies$ Player 2 wins positively, for that she needs to remember $\frac{1}{2}C^n_{n/2}$ sets among $C^n_{n/2}$.
- $\implies$ Player 2 needs doubly-exponential memory.

Deciding almost-sure, sure and positive winning

Theorem: [Bertrand, Genest, G.] there is an algorithm running in exponential time $(G \cdot 2^K)$ for deciding whether the game is winning positively for player 1 or surely for player 2.

Theorem: [Bertrand, Genest, G.] there is an algorithm running in doubly exponential time $(2^{G \cdot 2^K})$ for deciding whether the game is winning almost-surely for player 1 or positively for player 2.

Previous Work: [Chatterjee, Doyen, Henzinger, Raskin, 07] One player perfectly informed. No determinacy result. EXPTIME algorithm for deciding whether player 1 can win almost-surely a Büchi objective.

Lower bounds: EXPTIME and 2-EXPTIME complete.

Deciding almost-sure, sure and positive winning

Theorem: [Bertrand, Genest, G.] there is an algorithm running in doubly exponential time for deciding whether the game is winning almost-surely for player 1, surely for 2 or positively for both.

- Computation of smallest and greatest fix-points of monotonic operators.

Deciding almost-sure, sure and positive winning

Theorem: [Bertrand, Genest, G.] there is an algorithm running in doubly exponential time for deciding whether the game is winning almost-surely for player 1, surely for 2 or positively for both.

- Computation of smallest and greatest fix-points of monotonic operators.
- Deciding positive winning for player 1 or sure winning for player 2 in EXPTIME: operator $\mathcal{P}(\mathcal{P}(K)) \to \mathcal{P}(\mathcal{P}(K))$, each iteration computable in exponential time.

Deciding almost-sure, sure and positive winning

Theorem: [Bertrand, Genest, G.] there is an algorithm running in doubly exponential time for deciding whether the game is winning almost-surely for player 1, surely for 2 or positively for both.

- Computation of smallest and greatest fix-points of monotonic operators.
- Deciding positive winning for player 1 or sure winning for player 2 in EXPTIME: operator $\mathcal{P}(\mathcal{P}(K)) \to \mathcal{P}(\mathcal{P}(K))$, each iteration computable in exponential time.
- Deciding almost-surely winning for player 1 or positive winning for player 2 in 2-EXPTIME: operator $\mathcal{P}(K \times \mathcal{P}(K)) \to \mathcal{P}(K \times \mathcal{P}(K))$, iteration computable in doubly-exponential time.

Conclusion

- We solved qualitative questions about a very general model of reachability games with partial observation.

Conclusion

- We solved qualitative questions about a very general model of reachability games with partial observation.
- Better complexities in special cases.

Conclusion

- We solved qualitative questions about a very general model of reachability games with partial observation.
- Better complexities in special cases.
- Decidable questions about Büchi games.

Conclusion

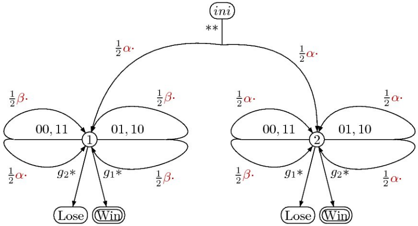
- We solved qualitative questions about a very general model of reachability games with partial observation.
- Better complexities in special cases.
- Decidable questions about Büchi games.
- Research directions:

Conclusion

- We solved qualitative questions about a very general model of reachability games with partial observation.
- Better complexities in special cases.
- Decidable questions about Büchi games.
- Research directions:
- Finding subclasses of games where the value is computable.

Conclusion

- We solved qualitative questions about a very general model of reachability games with partial observation.
- Better complexities in special cases.
- Decidable questions about Büchi games.
- Research directions:
- Finding subclasses of games where the value is computable.
- Deciding whether the value is 1.



Beliefs and pessimistic beliefs

- A signal c contains the action $i(c)$ that has just been played.

Beliefs and pessimistic beliefs

- A signal c contains the action $i(c)$ that has just been played.
- Definition: Let $L \subseteq K$ and c a signal. The belief of player 1 after L and signal c is:

  $$\mathcal{B}^1(L,c) = \{k \in S \mid \exists l \in L, \exists d \in D, p(k,c,d \mid l, i(c), j(d)) > 0\}.$$

  The pessimistic belief of player 1 after L and signal c is:

  $$\mathcal{B}^1_p(L,c) = \{k \in S \setminus T \mid \exists l \in L \setminus T, \exists d \in D, p(k,c,d \mid l, i(c), j(d)) >$$

Beliefs and pessimistic beliefs

- A signal c contains the action $i(c)$ that has just been played.
- Definition: Let $L \subseteq K$ and c a signal. The belief of player 1 after L and signal c is:

  $$\mathcal{B}^1(L,c) = \{k \in S \mid \exists l \in L, \exists d \in D, p(k,c,d \mid l, i(c), j(d)) > 0\}.$$

  The pessimistic belief of player 1 after L and signal c is:

  $$\mathcal{B}^1_p(L,c) = \{k \in S \backslash T \mid \exists l \in L \backslash T, \exists d \in D, p(k,c,d \mid l, i(c), j(d)) >$$

- Proposition: if player 1 plays an almost-surely winning strategy, his pessimistic beliefs are almost-surely winning. If player 2 plays a surely winning strategy, her beliefs are surely winning.

Player 2 has a belief strategy for winning surely

Proposition: For winning surely, player 2 has a strategy with finite memory of exponential size.

- Monotonic operator Φ on $\mathcal{P}(\mathcal{P}(S \setminus T))$ with $\Phi(\mathcal{L}) = \{L \in \mathcal{L} \mid \exists j_L \in J, \forall d \in D, (j = j(d)) \implies (\mathcal{B}(L, d) \in \mathcal{L})\}$.

Player 2 has a belief strategy for winning surely

Proposition: For winning surely, player 2 has a strategy with finite memory of exponential size.

- Monotonic operator Φ on $\mathcal{P}(\mathcal{P}(S \setminus T))$ with $\Phi(\mathcal{L}) = \{L \in \mathcal{L} \mid \exists j_L \in J, \forall d \in D, (j = j(d)) \implies (\mathcal{B}(L, d) \in \mathcal{L})\}$.
- Let $\mathcal{L}_\infty$ be the largest fix-point of Φ.

Player 2 has a belief strategy for winning surely

Proposition: For winning surely, player 2 has a strategy with finite memory of exponential size.

- Monotonic operator Φ on $\mathcal{P}(\mathcal{P}(S \setminus T))$ with $\Phi(\mathcal{L}) = \{L \in \mathcal{L} \mid \exists j_L \in J, \forall d \in D, (j = j(d)) \implies (\mathcal{B}(L, d) \in \mathcal{L})\}$.
- Let $\mathcal{L}_\infty$ be the largest fix-point of Φ.
- Strategy for player 2 consisting in playing actions j_L such that her belief L stays in $\mathcal{L}_\infty$.

Player 2 has a belief strategy for winning surely

Proposition: For winning surely, player 2 has a strategy with finite memory of exponential size.

- Monotonic operator Φ on $\mathcal{P}(\mathcal{P}(S \setminus T))$ with $\Phi(\mathcal{L}) = \{L \in \mathcal{L} \mid \exists j_L \in J, \forall d \in D, (j = j(d)) \implies (\mathcal{B}(L, d) \in \mathcal{L})\}$.
- Let $\mathcal{L}_\infty$ be the largest fix-point of Φ.
- Strategy for player 2 consisting in playing actions j_L such that her belief L stays in $\mathcal{L}_\infty$.
- Conversely, playing randomly, player 1 wins positively from any support in $\Phi^n(\mathcal{P}(K)) \setminus \Phi^{n+1}(\mathcal{P}(K))$.

Player 2 has a double exponential memory strategy for wining positively

Proposition: Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. Suppose player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L}$. Then,

(i) either every $L \notin \mathcal{L}$ is almost-surely winning for player 1,

Defines a monotonic operator on $\mathcal{P}(K)$.

Player 2 has a double exponential memory strategy for wining positively

Proposition: Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. Suppose player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L}$. Then,

(i) either every $L \notin \mathcal{L}$ is almost-surely winning for player 1,

(ii) or there exists $\mathcal{L}' \subseteq \mathcal{P}(K)$ such that

Defines a monotonic operator on $\mathcal{P}(K)$.

Player 2 has a double exponential memory strategy for wining positively

Proposition: Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. Suppose player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L}$. Then,

(i) either every $L \notin \mathcal{L}$ is almost-surely winning for player 1,

(ii) or there exists $\mathcal{L}' \subseteq \mathcal{P}(K)$ such that

  (a) $\mathcal{L}'$ is not empty and does not intersect $\mathcal{L}$,

Defines a monotonic operator on $\mathcal{P}(K)$.

Player 2 has a double exponential memory strategy for wining positively

Proposition: Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. Suppose player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L}$. Then,

(i) either every $L \notin \mathcal{L}$ is almost-surely winning for player 1,

(ii) or there exists $\mathcal{L}' \subseteq \mathcal{P}(K)$ such that

  (a) $\mathcal{L}'$ is not empty and does not intersect $\mathcal{L}$,

  (b) player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L} \cup \mathcal{L}'$,

Defines a monotonic operator on $\mathcal{P}(K)$.

Player 2 has a double exponential memory strategy for wining positively

Proposition: Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. Suppose player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L}$. Then,

(i) either every $L \notin \mathcal{L}$ is almost-surely winning for player 1,

(ii) or there exists $\mathcal{L}' \subseteq \mathcal{P}(K)$ such that

- (a) $\mathcal{L}'$ is not empty and does not intersect $\mathcal{L}$,
- (b) player 1 can enforce his pessimistic beliefs to stay outside $\mathcal{L} \cup \mathcal{L}'$,
- (c) player 2 has a strategy τ such that for every σ and initial distribution δ with support in $\mathcal{L}'$,

$$\mathbb{P}_\delta^{\sigma,\tau}\left(\forall n \in \mathbb{N}, K_n \notin T \mid \forall n \in \mathbb{N}, \mathcal{B}_p^1(L, C_1, \ldots, C_n) \notin \mathcal{L}\right) > 0 \ .$$

Defines a monotonic operator on $\mathcal{P}(K)$.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.
- Player 2 has the following positively winning strategy from $\mathcal{L}_\infty$.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.
- Player 2 has the following positively winning strategy from $\mathcal{L}_\infty$.
  - Play randomly for some time.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.
- Player 2 has the following positively winning strategy from $\mathcal{L}_\infty$.
  - Play randomly for some time.
  - Guess wildly the pessimistic belief $L \in \mathcal{L}_{n+1} \setminus \mathcal{L}_n$ of player 1.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.
- Player 2 has the following positively winning strategy from $\mathcal{L}_\infty$.
  - Play randomly for some time.
  - Guess wildly the pessimistic belief $L \in \mathcal{L}_{n+1} \setminus \mathcal{L}_n$ of player 1.
  - Win surely against player 1, under the hypothesis that his pessimistic belief will stay in $\mathcal{L}_{n+1} \setminus \mathcal{L}_n$.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.
- Player 2 has the following positively winning strategy from $\mathcal{L}_\infty$.
  - Play randomly for some time.
  - Guess wildly the pessimistic belief $L \in \mathcal{L}_{n+1} \setminus \mathcal{L}_n$ of player 1.
  - Win surely against player 1, under the hypothesis that his pessimistic belief will stay in $\mathcal{L}_{n+1} \setminus \mathcal{L}_n$.
  - Requires to make hypotheses about pessimistic beliefs of player 1, doubly-exponential size memory is required.

Positively winning strategy of player 2

- The collection $\mathcal{L}_\infty$ of positively winning supports for player 2 is computed as a smallest fix-point $\emptyset = \mathcal{L}_0 \subset \mathcal{L}_1 \subset \ldots \subset \mathcal{L}_\infty$.
- Player 2 has the following positively winning strategy from $\mathcal{L}_\infty$.
  - Play randomly for some time.
  - Guess wildly the pessimistic belief $L \in \mathcal{L}_{n+1} \setminus \mathcal{L}_n$ of player 1.
  - Win surely against player 1, under the hypothesis that his pessimistic belief will stay in $\mathcal{L}_{n+1} \setminus \mathcal{L}_n$.
  - Requires to make hypotheses about pessimistic beliefs of player 1, doubly-exponential size memory is required.
- For winning almost-surely from the complement of $\mathcal{L}_\infty$, player 1 plays randomly any action that ensures his pessimistic belief stays in the complement of $\mathcal{L}_\infty$.