

Outline

(1) A Model for Infinite Two-Player Games with Partial Information

- Two-Player Zero-Sum Games
- Partial Information

(2) Winning Strategies

(2.1) Powerset Construction

- Observation Based Winning Conditions
- Omega-Regular Winning Conditions

(2.2) Iterative Construction and Finite Memory

(2.3) Lower Bounds for the Memory

(2.4) Hiding Private Moves

(2.5) Lower Bounds for the Complexity

(2.6) Alternating Tree Automata

Two-Player Zero-Sum Games

A two-player zero-sum game has the form

$$G = (V, V_0, (f_a)_{a \in A}, W_0)$$

with

- $V \neq \emptyset$: set of positions
- V_0 : positions of player 0
- $A \neq \emptyset$: set of actions
- $f_a : \text{dom}(f_a) \subseteq V \rightarrow V$ for $a \in A$
- $W_0 \subseteq V(AV)^\omega$: winning condition for player 0

such that

$$\text{act}(v) := \{a \in A \mid v \in \text{dom}(f_a)\} \neq \emptyset \text{ for all } v \in V$$

Two-Player Zero-Sum Games

Plays and Strategies

Play from initial position $v \in V$:

Infinite sequence $\pi = v_0 a_0 v_1 a_1 v_2 \dots \in v(AV)^\omega$ such that $v_i \in \text{dom}(f_{a_i})$ and $f_{a_i}(v_i) = v_{i+1}$ for each $i < \omega$.

Strategy for player i :

Function $g : \{\pi \in P_{\text{fin}}(V) \mid \text{last}(\pi) \in V_i\} \rightarrow A$ such that $g(\pi) \in \text{act}(\text{last}(\pi))$ for all $\pi \in \text{dom}(g)$.

($P_{\text{fin}}(V)$: finite prefixes of plays in G from some $v \in V$.)

Prefix $\pi = v_0 a_0 v_1 \dots$ of a play in G is called compatible with g if for all $j < l(\pi)$ such that $v_j \in V_i$:

$$a_j = g(v_0 a_0 \dots a_{j-1} v_j).$$

Winning strategy from v :

Each play from v which is compatible with g is won by player i .

Two-Player Zero-Sum Games

Memory Strategies

Memory Structure:

$M = (S, \delta_0, \delta)$ with

- S : set of states
- $\delta_0 : V \rightarrow S$: initializing function
- $\delta : S \times (A \times V) \rightarrow S$: update function

Memory strategy for player i for G with respect to M :

Function $g : S \times V_i \rightarrow A$ such that

$g(s, v) \in \text{act}(v)$ for all $(s, v) \in S \times V_i$.

Prefix $\pi = v_0 a_0 v_1 \dots$ of a play in G from $v_0 \in V'$ is called compatible with g if for all $j < l(\pi)$ such that $v_j \in V_i$:

$a_j = g(\delta^*(v_0 a_0 v_1 \dots a_{j-1} v_j), v_j)$.

($\delta^* : P_{\text{fin}}(V) \rightarrow S$: usual inductive extension of δ .)

Partial Information

Example: Coin Game

- (C.1) Two piles of coins
- (C.2) Left pile has no coins, right pile has two coins
- (C.3) First, player 0 chooses to move zero or one coins from the right pile to the left
- (C.4) Then player 1 chooses to move zero or one coins from the right pile to the left
- (C.5) Player 0 wins, if at the end of the game, the piles have different height

Winning strategy **f** for player 1:

If after the move of player 0, the piles have equal height:
move zero coins

If after the move of player 0, the piles have different height:
move one coin

Partial Information

Example: Coin Game with Partial Information

(P.1) When player 0 moves, piles are hidden from player 1

(P.2) Player 1 has to decide on his move *before* the piles are shown to him again

\mathbf{f} is not a strategy in this setting anymore!

Possible strategies:

f_1 : Move zero coins

f_2 : Move one coin

Neither f_1 nor f_2 is a winning strategy for player 1.

Obviously:

Player 0 does not have a winning strategy as well.

So: The game is undetermined.

Notice: The game has a reachability objective.

Partial Information

The General Case

For $i \in \{0, 1\}$, define an equivalence relation \sim_i on the positions of the game tree:

$$u \sim_i v$$

$$:\iff$$

at u and v , player i has the *same information*.

Strategy:

$$\pi \sim_i \pi' \implies g(\pi) = g(\pi')$$

In the example:

Information of player 1 at $(0, 2)$ and at $(1, 1)$ is the same.

(All that he can derive from the fact that the initial position is $(0, 2)$ and that player 0 can move zero coins or one coins plus the fact that it is his turn.)

Partial Information

Finite Representation of Infinite Games

Finite representation of the game graph:

finite game graphs, pushdown systems, ...

Finite representation of the winning condition:

omega-automata, S1S, ...

Finite representation of the partial information: ?

Equivalence relations have to be defined on the unravelling of the game graph.

(For each finite play prefix we have to define the information of the players after this prefix is played.)

\rightsquigarrow Consider possibilities to define equivalence relations on finite play prefixes in a finite fashion.

Partial Information

Finite Automata

(A.1) Automata with output

- For each player i of the game consider a finite automaton \mathcal{A}_i with output function
- $\tau_i(\delta_i^*(\pi))$ is the information of player i after π has been played

(A.2) Automata over relations

- For each player i of the game consider a finite automaton \mathcal{A}_i over relations
- $\sim_i := L(\mathcal{A}_i)$
- Three possibilities:
 - (a) Componentwise recognizable relations
 - (b) Automatic relations
 - (c) Rational relations

Partial Information

Logical Formulas

$$\pi \sim_i \pi'$$

$$:\iff$$

$$\mathfrak{A} = (P_{\text{fin}}(V), \tau^{\mathfrak{A}}) \models \varphi(\pi, \pi')$$

for some formula $\varphi(x, y)$ with two free element variables x and y from some logical language \mathcal{L} over some appropriate signature τ .

Partial Information

Two Special Cases

(Step 1) Define the information that a player has about the positions and the actions in the game graph

positions: $\text{vis}_i^V : V \rightarrow \text{VIS}_i^V$

actions: $\text{vis}_i^A : A \rightarrow \text{VIS}_i^A$

(Step 2) Extend the resulting function $\text{vis}_i : AV \rightarrow \text{VIS}_i^A \text{VIS}_i^V$ to finite play prefixes

Extension:

Move $u \rightarrow v$ is private for player i , if $u \in V_i$ and $u \sim_{1-i}^V v$.

(*) $\text{vis}_i^*(\pi) = \text{vis}_i^V(\pi)$ if $\pi \in V$

(*) $\text{vis}_i^*(\pi av) = \text{vis}_i^*(\pi) \text{vis}_i^A(av)$.

(+) $\text{vis}_i^+(\pi) = \text{vis}_i^V(\pi)$ if $\pi \in V$.

(+) $\text{vis}_i^+(\pi av) = \text{vis}_i^+(\pi)$,

if $(\text{last}(\pi), v)$ is private for player $1 - i$.

(+) $\text{vis}_i^+(\pi av) = \text{vis}_i^+(\pi) \text{vis}_i^A(av)$, otherwise.

\rightsquigarrow Two different notions of strategies!

Partial Information

Two Special Cases

Automata:

(*) \sim_1^* is automatic but not componentwise recognizable

(+) \sim_1^+ is rational but not automatic

Logical Formulas: Consider the signatures

(*) $\tau^* = \{E, \sim\}$

(+) $\tau^+ = \{E, E_0, \sim\}$

and let

- $E^{\mathfrak{A}} = \{(\pi, \pi av) \mid \pi, \pi av \in P_{\text{fin}}(v_0)\}$.
- $E_0^{\mathfrak{A}} = \{(\pi, \pi av) \mid \text{last}(\pi) \in V_0 \text{ and } \text{last}(\pi) \sim_1^V v\}$.
- $\sim^{\mathfrak{A}} = \{(\pi, \pi') \mid \text{last}_2(\pi) \sim_1 \text{last}_2(\pi')\} \cup \{(v_0, v_0)\}$.

\sim_1^* and \sim_1^+ are

- Definable in LFP and GSO
- Not definable in L_μ^2 and MSO

Partial Information

Games With Partial Information

Two-player zero-sum game with partial information:

$$\mathcal{G} = (G, (\text{vis}_i^V)_{i=0,1}, (\text{vis}_i^A)_{i=0,1})$$

with

- $G = (V, V_0, (f_a)_{a \in A}, W_0)$: two-player zero-sum game
- For $i = 0, 1$, $\text{vis}_i^V : V \rightarrow \text{VIS}_i^V$ and $\text{vis}_i^A : A \rightarrow \text{VIS}_i^A$:
functions into sets VIS_i^V and VIS_i^A

such that

(C1) If $u, v \in V$ with $\text{vis}_i^V(u) = \text{vis}_i^V(v)$ then
 $u, v \in V_i$ or $u, v \notin V_i$

(C2) If $a, b \in A_i$ with $a \neq b$ then $\text{vis}_i^A(a) \neq \text{vis}_i^A(b)$

Partial Information

Strategies

Strategy for player i for \mathcal{G} [if private moves are hidden]:

Strategy g for player i for G such that $g(\pi) = g(\pi')$

for all $\pi, \pi' \in \text{dom}(g)$ with $\pi \sim_1^* \pi'$ [$\pi \sim_1^+ \pi'$].

Memory structure for player i for \mathcal{G}

[if private moves are hidden] :

Memory structure $M = (S, \delta_0, \delta)$ for G such that:

(1) $\delta_0(v) = \delta_0(w)$ for all $v, w \in V'$ with $v \sim_i^V w$

(2) $\delta(s, (a, v)) = \delta(s, (b, w))$ for all $s \in S$ and
all $(a, v), (b, w) \in A \times V$ with $Av \sim_i bw$

[(3) $\delta^*(\pi av) = \delta^*(\pi)$

if $(\text{last}(\pi), v)$ is a private move of player $1 - i$]

Memory strategy for player i for \mathcal{G} with respect to M :

Memory strategy $g : S \times V_i \rightarrow A$ for player i with respect
to M such that

for all $s \in S$ and all $u, v \in V$ with $u \sim_i^V v$:

$g(s, u) = g(s, v)$.

Winning Strategies

Strategy problem: [if private moves are hidden]

Given a (two-player, zero-sum) game \mathcal{G} with partial information and a position v ,

does player 1 have a winning strategy for \mathcal{G} from v ?

[if private moves are hidden]

Aim:

For large classes of games, find

- (efficient) algorithm for the strategy problem
- (efficient) method to implement winning strategies with (small) finite memory

Idea:

Turn a game with partial information into a game with full information such that the existence of winning strategies for player 1 is preserved.

\rightsquigarrow Powerset Construction

Powerset Construction

Originally suggested by John H. Reif in his paper
“The Complexity of Two-Player Games
of Incomplete Information”. (1984)

Limitations:

- Restricted game model
- Finitely branching game graphs
- Only reachability objectives

More recently extended to arbitrary observation based
winning conditions on finitely branching game graphs.
(Chatterjee, Doyen, Henzinger, Raskin:
Algorithms for Omega-Regular Games with Incomplete
Information. (2006))

Powerset Construction

Idea

- Resulting game is supposed to have full information
- \rightsquigarrow Both players always know the recent position of the game
- Existence of a winning strategy for player 1 from v_0 shall be preserved
- \rightsquigarrow Any position of the new game must capture all the uncertainties about the recent position that player 1 actually has after some $\pi \in P_{\text{fin}}(v_0)$ has been played (That means, it must contain all positions, that player i considers possible after π has been played)

So the positions of the new game are of the form:

$$v(\pi) = \{\text{last}(\pi') \mid \pi' \sim_i^* \pi\} \quad \text{for } \pi \in P_{\text{fin}}(v_0)$$

Powerset Construction

The Construction

$$\mathcal{G} = (G, (\text{vis}_i^V), (\text{vis}_i^A)) \text{ with } G = (V, V_0, (f_a)_{a \in A}, W_0)$$

game with partial information, $v_0 \in V$.

Corresponding game with full information:

$$\overline{G}_{v_0} = (\overline{V}, \overline{V}_0, (\overline{E}_a)_{a \in A}, \overline{W}_0)$$

with

- $\overline{V} = \{v(\pi) \mid \pi \in P_{\text{fin}}(v_0)\}$
- $\overline{V}_0 = \{v(\pi) \in \overline{V} \mid \text{last}(\pi) \in V_0\}$
- For $a \in A$, the edge relation \overline{E}_a is the union of:
 - $\{(v(\pi), v(\pi b v)) \mid b \sim_1^A a, v(\pi) \in \overline{V}_0\}$
 - $\{(v(\pi), v(\pi a v)) \mid a \in \bigcap_{v \in v(\pi)} \text{act}(v), v(\pi) \in \overline{V}_1\}$
- For $\overline{\pi} = \overline{v}_0 a_0 \overline{v}_1 a_1 \dots \in P(\overline{v}_0)$ with $\overline{v}_0 = \{v_0\}$, define:
 - $\overline{\pi} \in \overline{W}_1 : \iff$
 - for all $\pi = v_0 a'_0 v_1 a'_1 \dots \in P(v_0)$ with
 - $a'_i \sim_1^A a_i$ and $v_i \in \overline{v}_i$ for all $i < \omega$: $\pi \in W_1$.

Powerset Construction

Main Result

Theorem 1.

- (1) If there is a terminal position $\bar{v} \in \bar{V}_1$, then player 1 does not have a strategy for \mathcal{G} from v_0 .
- (2) If there is no terminal position $\bar{v} \in \bar{V}_1$, then player 1 has a winning strategy for \mathcal{G} from v_0 if and only if he has a winning strategy for \bar{G}_{v_0} from \bar{v}_0 .

Observation Based Winning Conditions

- For $\pi = v_0 a_0 v_1 a_1 \dots \in V(AV)^* \cup V(AV)^\omega$:
 $\text{obs}_1(\pi) := \text{vis}_1^V(v_0) \text{vis}_1^V(v_1) \dots$
- W_1 is called observation based, if
 for all $\pi = v_0 a_0 v_1 \dots, \pi' = w_0 b_0 w_1 \dots \in V(AV)^\omega$
 with $\text{obs}_1(\pi) = \text{obs}_1(\pi')$:
 $\pi \in W_1$ if and only if $\pi' \in W_1$
- W_1 observation based : $\pi \in W_1$ if and only if
 $\text{obs}_1(\pi) \in \text{obs}_1(W_1) := \{\text{obs}_1(\pi) \mid \pi \in W_1\}$
- For $S \subseteq V$ with $u \sim_1^V v$ for all $u, v \in S$:
 $\text{vis}_1^V(S) := \text{vis}_1^V(v)$ for some $v \in S$
- For $\bar{\pi} = \bar{v}_0 a_0 \bar{v}_1 a_1 \dots \in \bar{V}(\bar{A} \bar{V})^\omega$:
 $\text{obs}_1(\bar{\pi}) := \text{vis}_1^V(\bar{v}_0) \text{vis}_1^V(\bar{v}_1) \dots$

Proposition 1.

If for all $v \in V$ the set $\{f_a(v) \mid a \in \text{act}(v)\}$ is finite,

then for each $\bar{\pi} \in P(\bar{v}_0)$:

$\bar{\pi} \in \bar{W}_1$ if and only if $\text{obs}_1(\bar{\pi}) \in \text{obs}_1(W_1)$.

Observation Based Winning Conditions

A Special Case

Information compatible Muller-conditions

$$(\text{col}, \mathcal{F}_0), \mathcal{F}_0 \subseteq 2^C$$

on finitely branching game graphs.

(If $u \sim_1^V v$ then $\text{col}(u) = \text{col}(v)$.)

Winning condition of \overline{G}_{v_0} :

$(\overline{\text{col}}, \mathcal{F}_0)$ where

$$\overline{\text{col}} : \overline{V} \rightarrow C \quad \text{with} \quad \overline{\text{col}}(\overline{v}) = \text{col}(v) \text{ for some } v \in \overline{v}.$$

Omega-Regular Winning Conditions

$\mathcal{A} = (VA, Q, q_0, \Delta, \text{acc})$:

ω -automaton recognizing W_0 .

For each $\bar{\pi} = \bar{v}_0 a_0 \bar{v}_1 a_1 \dots \in P(\bar{v}_0)$:

$\bar{\pi} \in \overline{W}_0 \iff \exists \pi = v_0 a'_0 v_1 a'_1 \dots \in P(v_0)$

with $a'_i \sim_1^A a_i$ and $v_i \in \bar{v}_i$ for all $i < \omega$ such that $\pi \in W_0$.

Idea for automaton accepting \overline{W}_0 :

Guess such a π and at the same time simulate \mathcal{A} on π .

Formally:

$\mathcal{B} = (\overline{VA}, \overline{Q} = Q \times V, (q_0, v_0), \overline{\Delta}, \overline{\text{acc}})$ with

- $\overline{\text{acc}}$ in the obvious way
- $((p, v), \bar{v}a, (q, w)) \in \overline{\Delta} : \iff$
 $v \in \bar{v}$ and there is some action $b \sim_1^A a$ such that
 $b \in \text{act}(v)$, $f_b(v) = w$ and $(p, vb, q) \in \Delta$

Proposition 2.

For all $\bar{\pi} = \bar{v}_0 a_0 \bar{v}_1 a_1 \dots \in P(\bar{v}_0)$:

$\bar{\pi} \in \overline{W}_0$ if and only if $\bar{\pi} \in L(\mathcal{B})$.

Iterative Construction and Finite Memory

Aims:

- Construct \overline{G}_{v_0} from (\mathcal{G}, v_0) effectively
- Implement winning strategies with finite memory

Need:

Update mechanism which produces a set $v(\pi av)$ from the set $v(\pi)$ by a certain update rule which depends only on the set $v(\pi)$ and the pair (a, v) but not on the prefix π .

Proposition 3.

Let $\pi \in P_{\text{fin}}(v_0)$, $v \in V$ and $a \in A$ with $\pi av \in P_{\text{fin}}(v_0)$.

$$v(\pi av) = \text{Post}_{[a]_{\sim_1}}(v(\pi)) \cap [v]_{\sim_1}.$$

Iterative Construction and Finite Memory

Theorem 2.

Let \mathcal{G} be a finite game with partial information and omega-regular winning condition such that $\text{act}([u]_{\sim_1}) = \text{act}(u)$ for all $u \in V_1$.

Then for all $v_0 \in \text{Win}_1^{\mathcal{G}}$, player 1 has a memory winning strategy for \mathcal{G} from v_0 which uses only finitely many memory states. The corresponding memory structure and the memory winning strategy can be constructed effectively.

- Information compatible Muller-game:
 $2^{|V|} \cdot (|C|)!$ memory states
- Information compatible parity game:
 $2^{|V|}$ memory states

In both cases, the memory structure can be constructed in time exponential in $|V|$.

Iterative Construction and Finite Memory

Iterative Construction of \overline{G}_{v_0}

Define sets $V^i \subseteq 2^V$ for $i < \omega$ inductively:

- $V^0 = \{\{v_0\}\}$
- $V^{i+1} = \{\{v_0\}\} \cup \{\text{Post}_{[a]_{\sim_1}}(S) \cap [v]_{\sim_1} \mid S \in V^i, a \in A, v \in \text{Post}_a(S)\}$

For $a \in \overline{A}$ and $i < \omega$, $E_a^i \subseteq 2^V \times 2^V$ is the union of:

- $\{(S, \text{Post}_{[a]_{\sim_1}}(S) \cap [v]_{\sim_1}) \mid S \in V^i, v \in \text{Post}_{[a]_{\sim_1}}(S), S \subseteq V_0\}$
- $\{(S, \text{Post}_a(S) \cap [v]_{\sim_1}) \mid S \in V^i, v \in \text{Post}_a(S), a \in \text{act}(S), S \subseteq V_1\}$

Proposition 4.

- (1) $V^i = \{v(\pi) \mid \pi \in P_{\text{fin}}(v_0), l(\pi) \leq i + 1\}$.
- (2) $E_a^i = \{(\overline{v}, \overline{w}) \in \overline{E}_a \mid \overline{v} \in V^i\}$ for all $a \in \overline{A}$.

Iterative Construction and Finite Memory

- So: $\bar{V} = \bigcup\{V^i \mid i < \omega\}$
- Thus: $\bar{E}_a = \bigcup\{E_a^i \mid i < \omega\}$ for $a \in \bar{A}$
- \mathcal{G} finite \Rightarrow
 $\exists \bar{i} \leq 2^{|V|} : V^{\bar{i}} = V^j$ and $E_a^{\bar{i}} = E_a^j$ for all $j > \bar{i}$

Theorem 3.

Let \mathcal{G} be a finite information compatible Muller-game and let $v_0 \in V$. Then the corresponding Muller-game with full information can be constructed in time exponential in $|V|$.

Corollary 1.

The strategy problem for finite information compatible Muller-games can be solved in time exponential in $|V|$.

Lower Bounds for the Memory

Time Bounded Safety Games

Time bounded safety game with time bound $\alpha \leq \omega$:

$G = (V, V_0, (f_a)_{a \in A}, (R, \alpha))$ where

- $R \subseteq V$
- winning condition :

$$\pi = v_0 a_0 v_1 a_1 \dots \in W_0 \iff v_i \in R \text{ for all } i < \alpha$$

Proposition 5.

Let \mathcal{G} be a time bounded safety game with partial information and time bound α .

- (1) W_0 is ω -regular and can be recognized by a deterministic safety automaton with $\alpha + 2$ states.
- (2) G is uniformly positional determined.
- (3) If $v \in V$ such that player 1 has a winning strategy for \mathcal{G} from v , then he has a memory winning strategy for \mathcal{G} from v which uses at most $2^{|V|}$ states.
- (4) If \mathcal{G} is information compatible then for each $v_0 \in V$, \overline{G}_{v_0} is a time bounded safety game with time bound α and $\overline{R} = 2^R \cap \overline{V}$.

Lower Bounds for the Memory

Main Result

Theorem 4.

There is a sequence $(\mathcal{G}_n)_{n < \omega}$ of time bounded safety games with partial information and some initial position v_0 , such that for each $n < \omega$, the number of positions in \mathcal{G}_n and the time bound are linear in n and the following holds. Player 1 has a memory winning strategy for \mathcal{G}_n from v_0 which uses $2^n - 1$ memory states but he does not have such a strategy which uses at most $2^n - 2$ memory states.

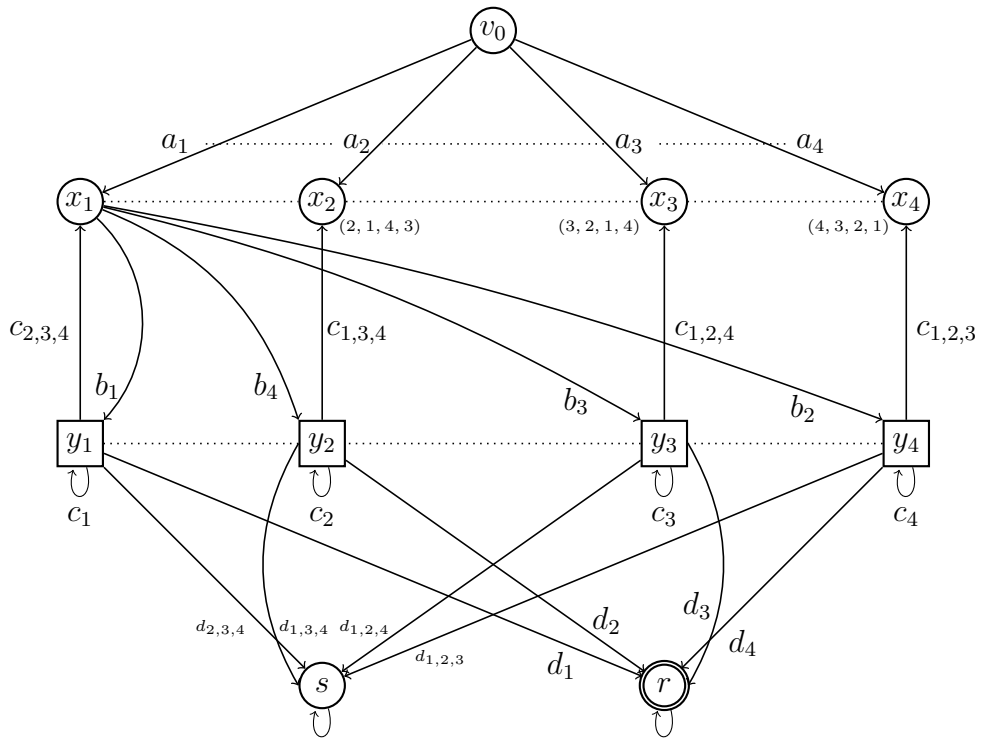
Similar idea:

Theorem 5.

There is a sequence $(\mathcal{G}_n)_{n < \omega}$ of safety games with partial information and some initial position v_0 , such that for each $n < \omega$, the number of positions in \mathcal{G}_n is quadratic in n and the following holds.

Player 1 has a memory winning strategy for \mathcal{G}_n from v_0 which uses $2^n - 1$ memory states but he does not have such a strategy which uses at most $2^n - 2$ memory states.

Lower Bounds for the Memory



Lower Bounds for the Memory

Conjecture 1. *For $1 < n < \omega$ let \mathcal{G}_n be the game as defined above without a time bound and let*

$$\{b_1, \dots, b_m\} = \{\sigma : N \rightarrow N \mid \sigma \text{ is bijective}\}.$$

Then each memory winning strategy for player 1 for \mathcal{G}_n from v_0 uses at least $2^n - 1$ memory states.

Hiding Private Moves

General Case

(1) Adaption of powerset construction

- Transforms omega-regular winning conditions into omega-regular winning conditions.
- Omega-automaton for \overline{W}_0 can be constructed effectively from omega-automaton for W_0 .

(2) Adaption of iterative construction

\rightsquigarrow

The strategy problem for ω -regular games where private moves are hidden is decidable. Finite memory strategies suffice to win and can be constructed effectively.

Information Compatible Muller-games

Polynomial time reduction of the strategy problem where private moves are hidden to the corresponding strategy problem.

Lower Bounds for the Complexity

Theorem 6.

The strategy problem for reachability games with partial information is EXPTIME-complete.

Proof-idea:

- Consider $L \in \text{EXPTIME} = \text{APSPACE}$
- \rightsquigarrow One-tape ATM M with $L(M) = L$
- For input w , \exists guesses an accepting run of M on w letter by letter
- \forall checks correctness of run
 - For each configuration: choose a letter to store
 - Store number of letter
 - Check corresponding letter in next configuration
- Actions of \forall are hidden from \exists
- Space bound: logarithmic

Alternating Tree Automata

Nonemptiness for nondeterministic tree automaton \mathcal{A} :

$L(\mathcal{A}) \neq \emptyset \iff$ exists tree t : exists run ρ of \mathcal{A} on t :

all infinite paths through ρ are accepting.

Game:

- Player \exists : Chooses tree and run
(by choosing transitions)
- Player \forall : Chooses path
(by choosing directions in the input tree
= directions in the run)

$L(\mathcal{A}) \neq \emptyset \iff$ Player \exists wins the game.

Alternating tree automaton:

Directions in the input tree \neq directions in the run.

(Several directions in the run may correspond to one direction in the tree.)

Labelling of the input tree may depend on the directions of the input tree that \forall chooses but it must not depend on the directions of the run that \forall chooses.

Alternating Tree Automata

Idea:

Split \exists into players T , guessing the tree and A , guessing the run of the automaton.

\leadsto Three player game with partial information.

- Players \forall and A have full information
- Player T sees only the branches of the input tree which are chosen

$L(\mathcal{A}) \neq \emptyset$ if and only if T and A can *cooperate* to win.

If \mathcal{A} is universal, then the game is a two-player game with partial information!

Alternating Tree Automata

Vice Versa

Problem:

Given three-player game with partial information where only player 1 has partial information, position v .

Can player 1 and 2 cooperate to win from v ?

First step:

Construct nondeterministic tree automaton such that a tree is accepted if and only if it is the unravelling of the game graph from v with:

The labellings at the positions of player 1 define a (full information) strategy f for player 1, such that there is a strategy g for player 2 with:

The composition of f and g is winning.

Second step:

Restrict the strategies of player 1 to partial information strategies.

Technique: “Narrowing”

(Kupferman, Vardi: “Church’s Problem Revisited”. (’99))

Alternating Tree Automata

If the game is a two-player game:

- the automaton from the first step is deterministic
- the “narrowing” of a deterministic automaton is universal