INFINITE GAMES

AMAN RAJ (170101006)

MAYANK WADHWANI (170101038)

EXAMPLE

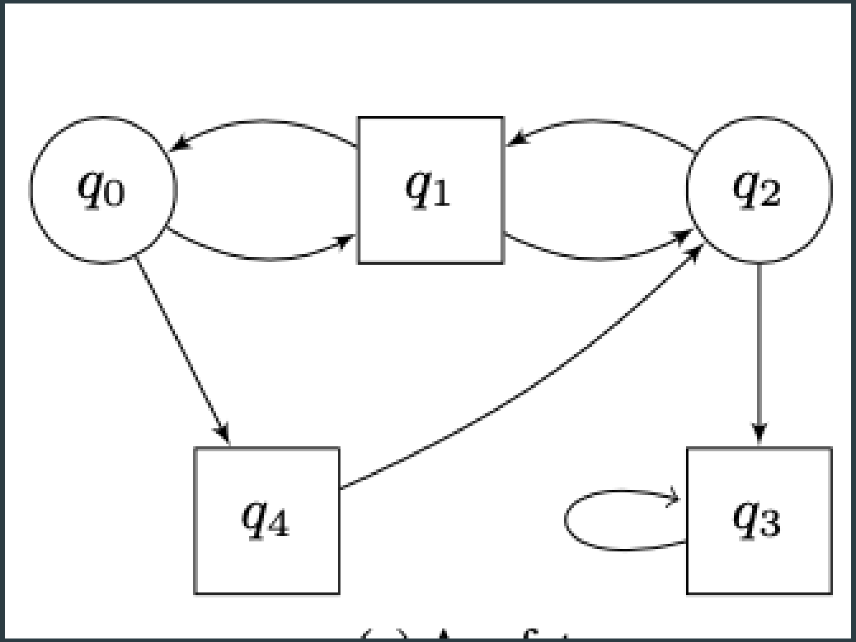
	C	D
A	$(0, 2)$	$(1, 1)$
B	$(1, 1)$	$(1, 2)$

- Dominated strategy:
  A strategy of a player dominates another one if the outcome of the first strategy is better
- Strategy B of player 1 dominates strategy A
- If player 2 knows that player 1 prefers strategy B to strategy A, then he will in turn prefer D to C
- (B, D) will be played
- This is iterated elimination of dominated strategies
- Surviving strategies are called iteratively admissible strategies.

BASIC TERMS

- MULTIPLAYER GAMES

$$\mathcal{G} = \langle P, (V_i)_{i \in P}, E, (\mathrm{WIN}_i)_{i \in P} \rangle$$



BASIC TERMS

- MULTIPLAYER GAMES
- HISTORY AND RUN |ã| = n, if n is finite -> history else run

BASIC TERMS

- MULTIPLAYER GAMES
- HISTORY AND RUN |ã| = n, if n is finite -> history else run
- Occ(path) $\{s \in V \mid \exists i \in \mathbb{N}.\ i < |\rho|, \rho_i = s\}.$

BASIC TERMS

- MULTIPLAYER GAMES
- HISTORY AND RUN |ã| = n, if n is finite -> history else run
- Occ(path)
- Stratergies (a function ->

$$(V^* \cdot V_i) \to V, \textit{ such that if } \sigma_i(\rho) = s \textit{ then } (\mathrm{last}(\rho), s) \in E.$$

BASIC TERMS

- MULTIPLAYER GAMES
- HISTORY AND RUN |ã| = n, if n is finite -> history else run
- Occ(path)
- Stratergies
- 1) S = set of all stratergies
- 2) S(i) = set of stratergies for state I
- 3) S(-i) = set of stratergies for all states except i

BASIC TERMS

- MULTIPLAYER GAMES
- HISTORY AND RUN |ã| = n, if n is finite -> history else run
- Occ(path)
- Stratergies
- Winning Stratergies

A strategy σ_i of player i is said to be *winning from state s against a rectangular set* $S_{-i} \subseteq \mathcal{S}_{-i}$, if for all $\sigma_{-i} \in S_{-i}$, $Out_s(\sigma_i, \sigma_{-i}) \in \mathrm{WIN}_i$. It is simply said *winning from state s* if $S_{-i} = \mathcal{S}_{-i}$. For each player i, we write $\mathrm{WIN}_i^s(\sigma_P)$ if $Out_s(\sigma_P) \in \mathrm{WIN}_i$.

SOME WINNING CONDITIONS REVISITED

- A *safety condition* is defined by a set $Bad_i \subseteq V$: $\textsc{Win}_i = (V \setminus Bad_i)^\omega$.

- A *reachability condition* is defined by a set $Good_i \subseteq V$: $\textsc{Win}_i = V^* \cdot Good_i \cdot V^\omega$.

- A *Büchi condition* is defined by a set $F_i \subseteq V$: $\textsc{Win}_i = (V^* \cdot F_i)^\omega$.

ADMISSIBILITY

- σ very weakly dominates strategy σ' with respect to S:

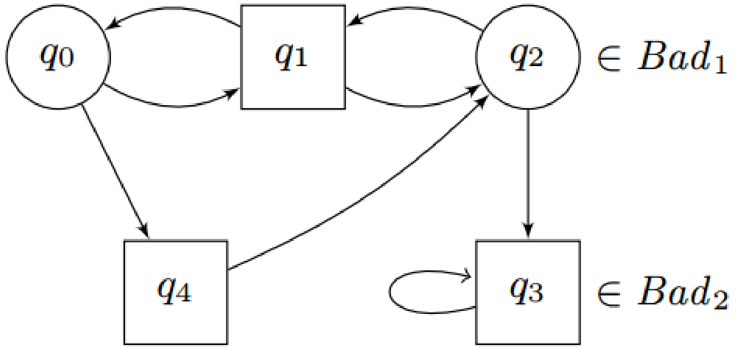
$$\forall \tau \in S_{-i}, \mathrm{WIN}_i^s(\sigma', \tau) \Rightarrow \mathrm{WIN}_i^s(\sigma, \tau)$$

- σ weakly dominates strategy σ' with respect to S, if σ very weakly dominates strategy σ' but vice versa is false.
- A strategy $\sigma \in S_i$ is dominated in S if there exists $\sigma' \in Si$ such that σ' weakly dominates σ
- A strategy that is not dominated in S is admissible in S.

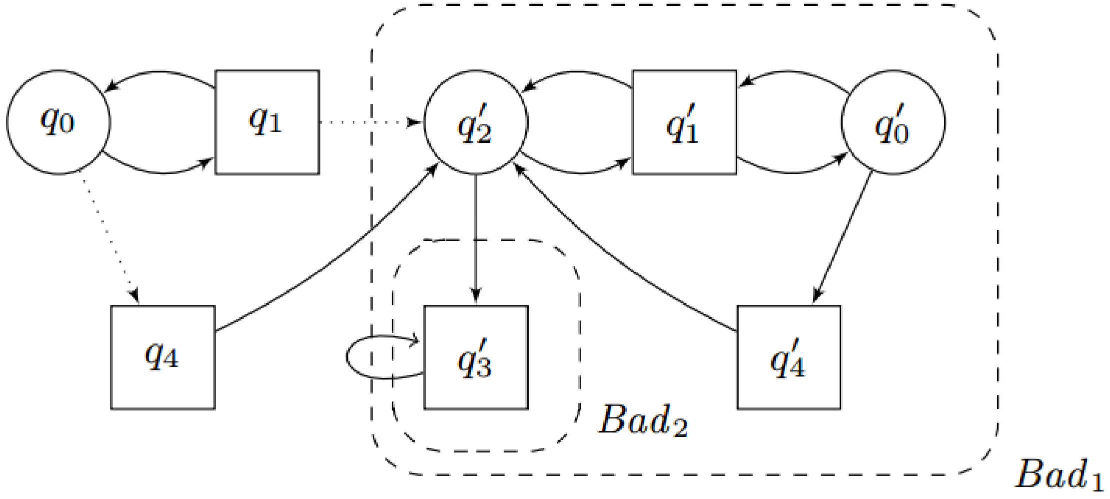
ADMISSIBILITY (CONT.)

- The set S* of iteratively admissible strategies is obtained by iteratively eliminating dominated strategies, starting from set S
- • $\mathcal{S}^0 = \mathcal{S}$;
  • $\mathcal{S}^{n+1} = \prod_{i \in P} \{\sigma_i \in \mathcal{S}_i^n \mid \sigma_i \text{ admissible in } \mathcal{S}^n\}$
- $\mathcal{S}^* = \bigcap_{n \in \mathbb{N}} \mathcal{S}^n$

EXAMPLE



(a) A safety game.



VALUE

- Our algorithms are based on the notion of value of a history
- It characterizes whether a player can win (alone) or cannot win (even with the help of other players)
- The value of history h for player i after the n-th step of elimination, written as Val_i^n (h).
- No strategy profile Sn winning for player i, then value = -1
- If a strategy is always winning for player i, then value = 1
- Else value = 0

SOME MORE TERMS

- The winning coalition problem
- Given a set of vertices, 2 subsets W,L, find an iteratively admissible profile such that all players in W wins and all players In L lose the game.
- Our robot motion is a type of winning coalition problem

SOME MORE TERMS

- The winning coalition problem
- lambda

$$\lambda(h) = \{i \in P \mid \exists k < |h|,\ h_k \in Bad_i\}.$$

SOME MORE TERMS

- The winning coalition problem
- Lambda
- T -> basically set of all badtransitions (all transitions where if we take that transition, we will lose)

Definition 5. *We write T_i^n for the set of transitions $s \to s' \in E$, such that s is controlled by player i and $\mathrm{Val}_i^n(s) > \mathrm{Val}_i^n(s')$. Such transitions are said to be* dominated *after the n-th step of elimination. We write T^n for the union of all T_i^n.*

SOME MORE TERMS

- The winning coalition problem
- Lambda
- T -> basically set of all badtransitions (all transitions where if we take that transition, we will lose)
- Doubt -> we wont have a transition from 1->-1 right?

SOME MORE TERMS

- The winning coalition problem
- Lambda
- T -> basically set of all useless transitions (all transitions where if we take that transition, we will lose)
- Subgame

Definition 6 (Subgame). *Let* $\mathcal{G} = \langle P, V, E, \textsc{Win}_P \rangle$ *be a game and* $T \subseteq E$ *a set of transitions. If each state* $s \in V$ *has at least one successor by* $E \setminus T$, *the game* $\mathcal{G} \setminus T = \langle P, V, E \setminus T, \textsc{Win}_P \rangle$ *is*

PROPOSITIONS

- Lemma 1:
  A player that plays according to an admissible strategy cannot go to a state that changes the value of the current history
- Players losing on h: $\lambda(h) = \{i \in P \mid \exists k < |h|,\ h_k \in Bad_i\}$
- Proposition 1:
  For safety winning conditions, the value of a history h only depends on λ(h) and last(h)
- We encode the set λ(h) of losing players in the state of the game
- New game:
  States in $2^P \times V$
  Set of transitions $(\lambda, s) \to (\lambda \cup \{i \mid s' \in Bad_i\}, s')$ for any λ ⊆ P, if s -> s'

DOMINANCE OF TRANSITIONS

- Definition:
  T_i^n for the set of transitions s -> s' ∈ E, s.t. s is controlled by player i and
  Val_i^n (s) > Val_i^n (s')
- Such transitions are said to be dominated after the n-th step of elimination
- T^n for the union of all T_i^n

- Subgame:
  Let G = <P, V, E, WIN_P> be a game and T ⊆ E a set of transitions.
  If each state s ∈ V has at least one successor by E \ T ,
  the game G \ T = <P, V, E \ T, WIN_P> is called a subgame of G

- Proposition 2:
  All admissible strategies w.r.t. S^n of player i are strategies of S_i (G \ T_i^n)

ALGORITHM

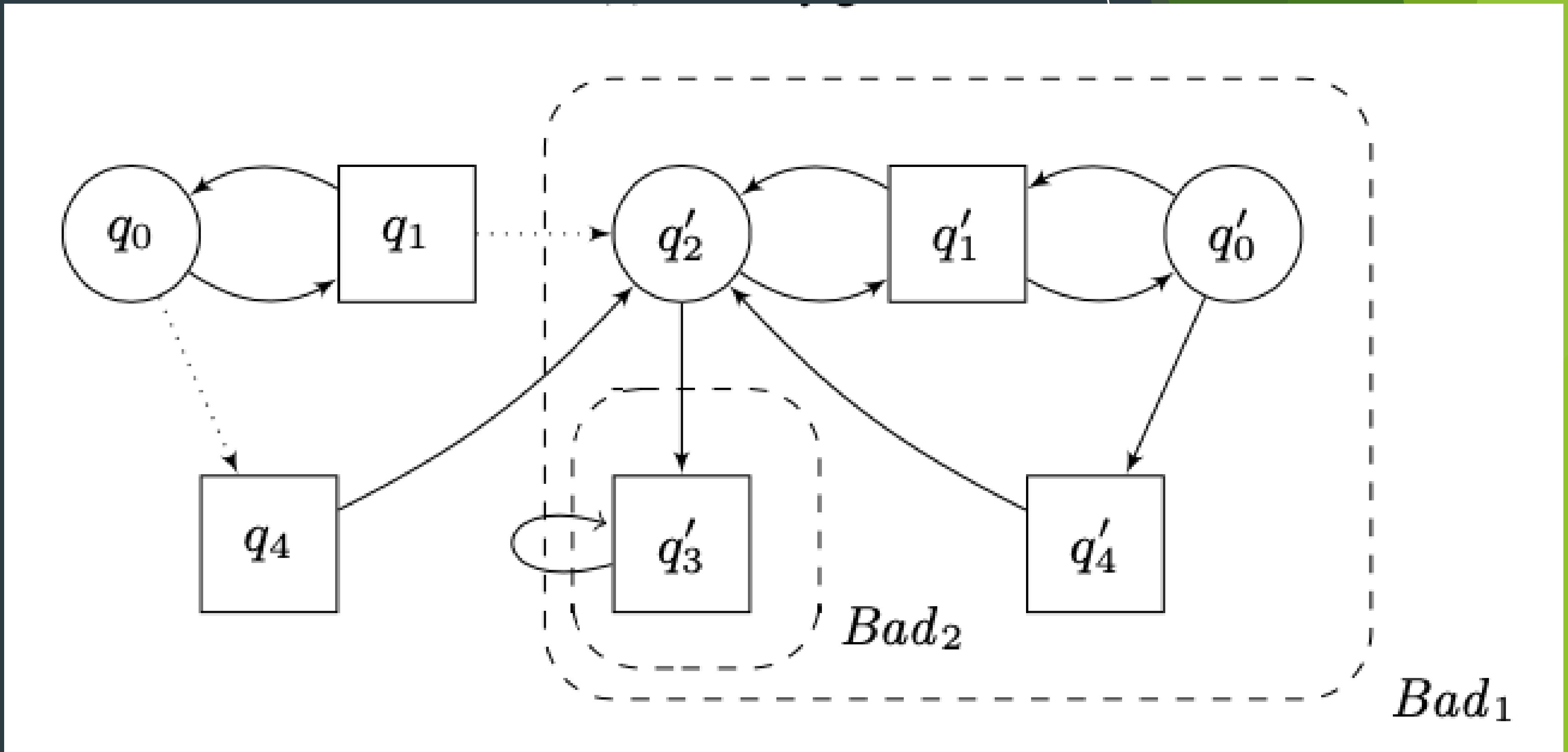
Algorithm 1: Computing the set of iteratively admissible strategies

$n := 0$; $T_i^{-1} := \emptyset$;
repeat
  forall the $s \in V$ **do**
    if *there is a winning strategy for player i from s in* $\mathcal{G} \setminus T^{n-1}$ **then** $\mathrm{Val}_i^n(s) := 1$;
    else if *there is no winning run for player i from s in* $\mathcal{G} \setminus T^{n-1}$ **then** $\mathrm{Val}_i^n(s) := -1$;
    else $\mathrm{Val}_i^n(s) := 0$;
  forall the $i \in P$ **do**
    $T_i^n := T_i^{n-1} \cup \{(s, s') \in E \mid s \in V_i \wedge \mathrm{Val}_i^n(s) > \mathrm{Val}_i^n(s')\}$;
  $n := n + 1$;
until $\forall i \in P.\ T_i^n = T_i^{n-1}$;

EXAMPLE REVISITED

- Val of q4 is –1 initially
- So q0->q4 removed

- Now, player 2 may lose if go to q2', so remove that edge



PROBLEM STATEMENT

- ROBOT MOTION PROBLEM

The game structure $\mathcal{G}^{\mathcal{M}}$ of imperfect information corresponding to multi-agent system $\mathcal{M} = \{R_1, R_2\}$ is a tuple $\mathcal{G}^{\mathcal{M}} = (\mathcal{V}, \Lambda, \tau, \mathcal{OBS}, \gamma)$ where

$$\mathcal{V} = \{t\} \cup \mathcal{O}_1 \cup \mathcal{O}_2,$$

$$\Lambda = \Lambda_1 \cup \Lambda_2,$$

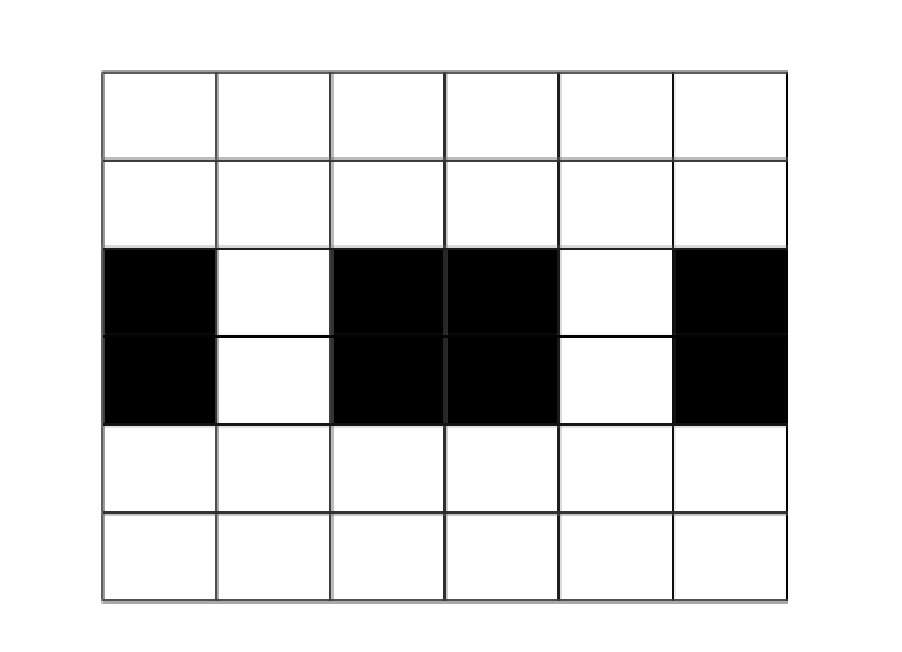
$$\tau = \tau_e \vee \tau_s,$$

$$\tau_e = t = 1 \wedge t' = 2 \wedge \tau_1 \wedge Same(\mathcal{O}_2, \mathcal{O}_2'),$$

$$\tau_s = t = 2 \wedge t' = 1 \wedge \tau_2 \wedge Same(\mathcal{O}_1, \mathcal{O}_1'),$$

$$\mathcal{OBS} = \mathcal{OBS}_2, \textit{ and}$$

$$\gamma = \gamma_2. \qquad \square$$



PLAN OF ACTION

- ROBOT MOTION PROBLEM
- Define -> initial predicate, global predicate
- In some way, convert our game to a safety game
- Find admissible stratergies for this safety game
- Composition
- Concurrency??