Different Strokes in Randomised Strategies: Revisiting Kuhn's Theorem Under Finite-memory Assumptions

James C. A. Main   Mickael Randour

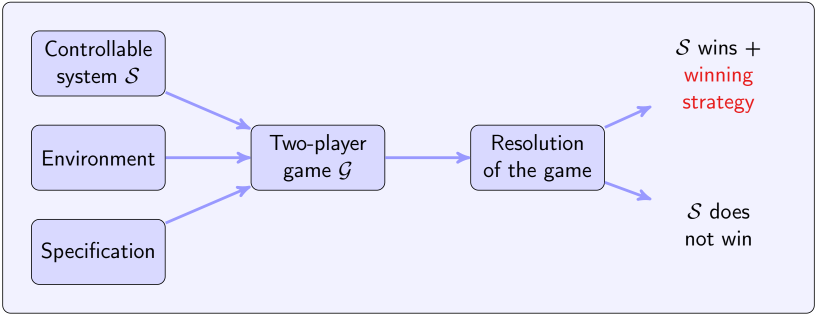
UMONS – Université de Mons and F.R.S.-FNRS

PhD Day – 13 May 2022

Table of contents

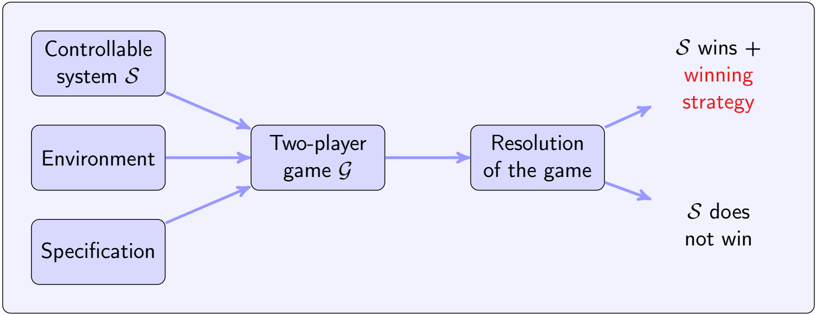
Reactive synthesis

- Reactive synthesis: automated generation of a controller for a reactive system from a specification with formal guarantees on the behaviour of the system.



Reactive synthesis

- Reactive synthesis: automated generation of a controller for a reactive system from a specification with formal guarantees on the behaviour of the system.



- A strategy is a formal blueprint of the sought controller for the system $\rightsquigarrow$ we need a finite implementation.

Kuhn's theorem

In this talk, we discuss randomised strategies. In general, one can define randomised strategies in different ways.

- Mixed strategies randomise between pure strategies at the start.
- Behavioural strategies randomly select an action at each step.

In general, these two classes of strategies are not comparable.

Kuhn's theorem

In this talk, we discuss randomised strategies. In general, one can define randomised strategies in different ways.

- Mixed strategies randomise between pure strategies at the start.
- Behavioural strategies randomly select an action at each step.

In general, these two classes of strategies are not comparable.

Kuhn's theorem

In games of perfect recall any mixed strategy has an equivalent behavioural strategy and vice-versa.

Kuhn's theorem

In this talk, we discuss randomised strategies. In general, one can define randomised strategies in different ways.

- Mixed strategies randomise between pure strategies at the start.
- Behavioural strategies randomly select an action at each step.

In general, these two classes of strategies are not comparable.

Kuhn's theorem

In games of perfect recall any mixed strategy has an equivalent behavioural strategy and vice-versa.

There exist different definitions of randomised finite-memory strategies.

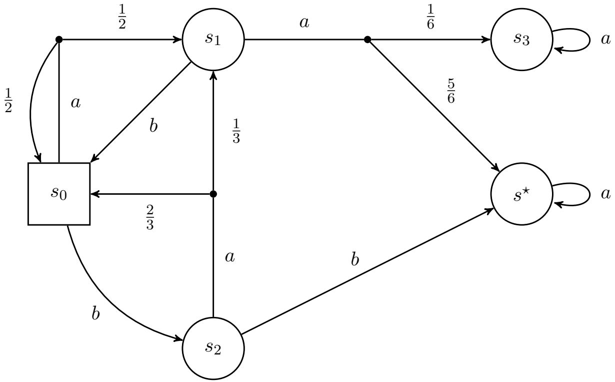
Main contribution

An adaptation of Kuhn's theorem for finite-memory strategies.

Stochastic games on graphs

Example

We consider two-player stochastic games of perfect information.



Stochastic games on graphs

Definition

Definition

A stochastic game of perfect information is a tuple $\mathcal{G} = (S_1, S_2, A, \delta)$ where

- $S = S_1 \uplus S_2$ is a finite set of states, S_i is the set of $\mathcal{P}_i$ states;
- A is a finite set of actions;
- $\delta \colon S \times A \to \mathcal{D}(S)$ is a partial transition relation.

For all $s \in S$, let $A(s) = \{a \in A \mid \delta(s, a) \text{ is defined}\}$ denote the set of actions enabled in s. We assume that in each state $s \in S$, there is at least one enabled action.

Stochastic games on graphs

Definition

Definition

A stochastic game of perfect information is a tuple $\mathcal{G} = (S_1, S_2, A, \delta)$ where

- $S = S_1 \uplus S_2$ is a finite set of states, S_i is the set of $\mathcal{P}_i$ states;
- A is a finite set of actions;
- $\delta \colon S \times A \to \mathcal{D}(S)$ is a partial transition relation.

For all $s \in S$, let $A(s) = \{a \in A \mid \delta(s, a) \text{ is defined}\}$ denote the set of actions enabled in s. We assume that in each state $s \in S$, there is at least one enabled action.

- Play: sequence $s_0 a_0 s_1 \ldots$ where for all $k \in \mathbb{N}$, $a_k \in A(s_k)$ and $\delta(s_k, a_k)(s_{k+1}) > 0$.
- History: prefix of a play ending in a state. We write $\mathsf{Hist}_i(\mathcal{G})$ for the set of histories ending in S_i.

Table of contents

Strategies: a formal definition

In general, a strategy of $\mathcal{P}_i$ provides a distribution over actions at each step of the play controlled by $\mathcal{P}_i$.

Definition

Let $i \in \{1, 2\}$. A strategy of $\mathcal{P}_i$ is a function $\sigma \colon \mathsf{Hist}_i(\mathcal{G}) \to \mathcal{D}(A)$ such that for all $h = s_0 a_1 s_1 \dots s_n \in \mathsf{Hist}_i(\mathcal{G})$, and all $a \in A$,

$$\sigma(h)(a) > 0 \implies a \in A(s_n).$$

- Strategies as defined above are behavioural strategies.

Comparing strategies

Given an initial state $s_{\mathsf{init}} \in S$, strategies σ_1 and σ_2 of $\mathcal{P}_1$ and $\mathcal{P}_2$ induce a probability distribution over plays, denoted $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}$, such that for any history $h = s_0 a_0 \dots a_{n-1} s_n$ with $s_0 = s_{\mathsf{init}}$, we have

$$\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}}(\mathsf{Cyl}(h)) = \prod_{k=0}^{n-1} \sigma_{i(k)}(s_0 a_0 \dots s_k)(a_k) \cdot \delta(s_k, a_k, s_{k+1}),$$

where $i(k) = 1$ if $s_k \in S_1$ and $i(k) = 2$ otherwise, and $\mathsf{Cyl}(h)$ denotes the set of plays that have h as a prefix.

Outcome-equivalence of strategies

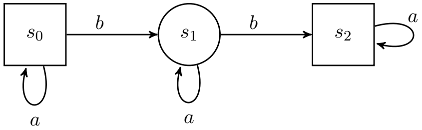
Two strategies σ_1 and τ_1 of $\mathcal{P}_1$ are outcome-equivalent if for all strategies σ_2 of $\mathcal{P}_2$ and all initial states $s_{\mathsf{init}} \in S$, $\mathbb{P}^{\sigma_1,\sigma_2}_{s_{\mathsf{init}}} = \mathbb{P}^{\tau_1,\sigma_2}_{s_{\mathsf{init}}}$.

Some strategies may not be implemented

- In some cases, strategies may require infinite memory to win.

Some strategies may not be implemented

- In some cases, strategies may require infinite memory to win.



- Objective: $\{(s_0a)^\omega\} \cup \{(s_0a)^k s_0 b (s_1 a)^k s_1 b (s_2 a)^\omega \mid k \in \mathbb{N}\}$.
- A winning strategy needs to count the number of occurrences of s_0 at the start of the play: requires an unbounded counter.

Finite-memory strategies

The classical model of finite-memory strategies is based on automata.

Definition

A $\mathcal{P}_i$ strategy is finite-memory if there is a Mealy machine $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{up}}, \alpha_{\mathsf{act}})$ that encodes it, where:

- M is a finite set of states;
- $\mu_{\mathsf{init}} \in \mathcal{D}(M)$ is an initial distribution;
- $\alpha_{\mathsf{up}} \colon M \times S \times A \to \mathcal{D}(M)$ is a memory-update function;
- $\alpha_{\mathsf{act}} \colon M \times S_i \to \mathcal{D}(A)$ is a next-move function.

Finite-memory strategies

The classical model of finite-memory strategies is based on automata.

Definition

A $\mathcal{P}_i$ strategy is finite-memory if there is a Mealy machine $\mathcal{M} = (M, \mu_{\mathsf{init}}, \alpha_{\mathsf{up}}, \alpha_{\mathsf{act}})$ that encodes it, where:

- M is a finite set of states;
- $\mu_{\mathsf{init}} \in \mathcal{D}(M)$ is an initial distribution;
- $\alpha_{\mathsf{up}} \colon M \times S \times A \to \mathcal{D}(M)$ is a memory-update function;
- $\alpha_{\mathsf{act}} \colon M \times S_i \to \mathcal{D}(A)$ is a next-move function.

The flow is as follows:

- An initial memory state m_0 is drawn following μ_{init}.
- If the state $s_n \in S_i$, the action a_n is drawn from $\alpha_{\mathsf{next}}(m_n, s_n)$.
- The next memory state m_{n+1} is drawn from $\alpha_{\mathsf{up}}(m_n, s_n, a_n)$.

Table of contents

Restricting randomisation in Mealy machines

- In the literature of formal methods, often the variant where only the output of the Mealy machines are randomised is considered.

$\rightsquigarrow$ Do we have an equivalent of Kuhn's theorem between this restricted class of Mealy machines and general ones ?

Restricting randomisation in Mealy machines

- In the literature of formal methods, often the variant where only the output of the Mealy machines are randomised is considered.

$\rightsquigarrow$ Do we have an equivalent of Kuhn's theorem between this restricted class of Mealy machines and general ones ?

- The answer is no: the smaller class is strictly less expressive.

Restricting randomisation in Mealy machines

- In the literature of formal methods, often the variant where only the output of the Mealy machines are randomised is considered.

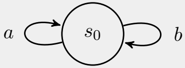
⇝ Do we have an equivalent of Kuhn's theorem between this restricted class of Mealy machines and general ones ?

- The answer is no: the smaller class is strictly less expressive.

Only randomised outputs vs. only randomised initialisation

In the game depicted below:

- randomised outputs can induce infinitely many plays;
- randomised initialisation can only induce finitely many.



Classifying Mealy machines

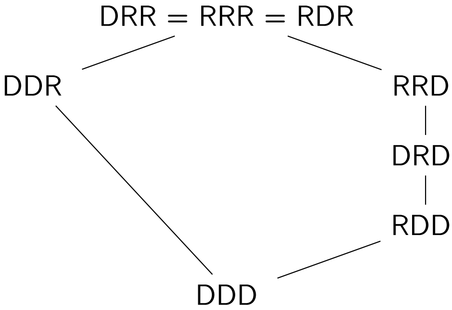
We use acronyms to define classes of Mealy machines: we use XYZ where X, Y, Z$\in \{$D, R$\}$ where D stands for deterministic and R for random, and

- X characterises initialisation,
- Y characterises outputs (next-move function),
- Z characterises updates.

Classifying Mealy machines

We use acronyms to define classes of Mealy machines: we use XYZ where X, Y, Z$\in$ {D, R} where D stands for deterministic and R for random, and

- X characterises initialisation,
- Y characterises outputs (next-move function),
- Z characterises updates.



References I

Aumann, Robert J . "28. Mixed and Behavior Strategies in Infinite Extensive Games". In: *Advances in Game Theory. (AM-52), Volume 52*. Princeton University Press, 2016, pp. 627–650. DOI: doi:10.1515/9781400882014-029. URL: https://doi.org/10.1515/9781400882014-029.