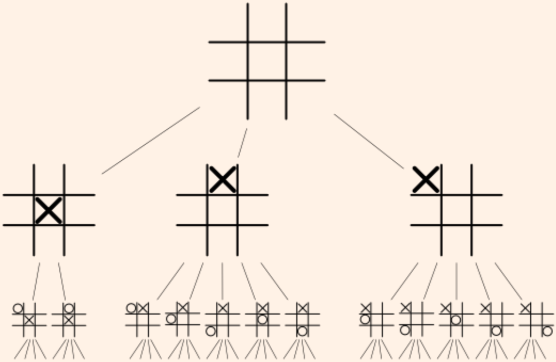
Simplifying Imperfect Recall Games

Hugo Gimbert LaBRI, CNRS, University of Bordeaux
Soumyajit Paul University of Liverpool
B. Srivathsan Chennai Mathematical Institute

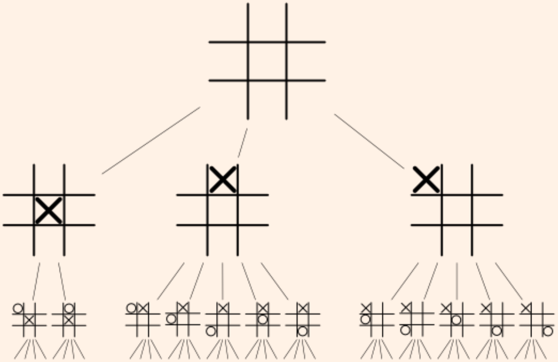
DETROIT
aamas2025

Perfect information games



State space of Tic Tac Toe

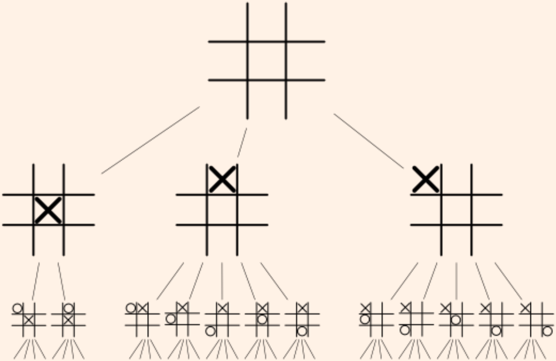
Perfect information games



State space of Tic Tac Toe

Players know their exact position in the state space

Perfect information games



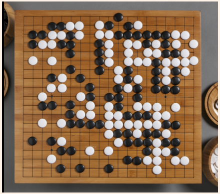
State space of Tic Tac Toe

Players know their exact position in the state space

Perfect information games can be solved in PTIME using bottom-up traversal. [Zermelo'1913]

Few more perfect information games...

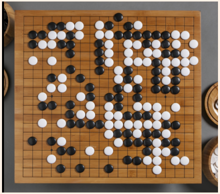
Chess



Go

Few more perfect information games...

Chess



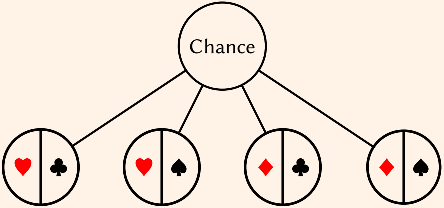
Go

Next : Games with imperfect information

Toy Card Game

Toy Card Game

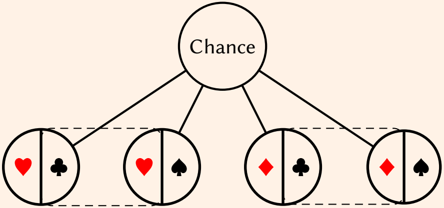
Players **Red** and **Black**, each gets a suit privately from ♥, ♣, ♦, ♠ matching player's color. **Red** plays first.



State space

Toy Card Game

Players **Red** and **Black**, each gets a suit privately from ♥, ♣, ♦, ♠ matching player's color. **Red** plays first.

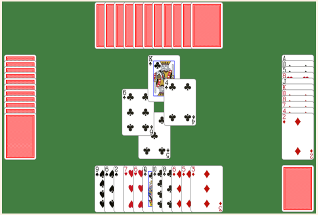


State space

Imperfect information : same knowledge in some states

Some games with imperfect information…

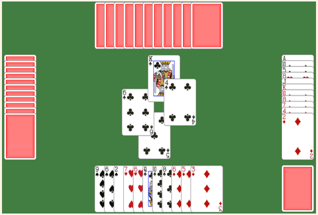
Poker



Bridge

Some games with imperfect information...

Poker



Bridge

Can we solve imperfect information games efficiently?

Solving imperfect information games is NP-hard [Koller, Megiddo'92]

Solving imperfect information games is NP-hard [Koller, Megiddo'92]

There are "simple" classes of imperfect information games solvable in PTIME

Solving imperfect information games is NP-hard [Koller, Megiddo'92]

There are "simple" classes of imperfect information games solvable in PTIME

Our contributions in this work

Solving imperfect information games is NP-hard [Koller, Megiddo'92]

There are "simple" classes of imperfect information games solvable in PTIME

Our contributions in this work

♠ New PTIME solvable class of imperfect information games by transformation to equivalent simpler games

Solving imperfect information games is NP-hard [Koller, Megiddo'92]

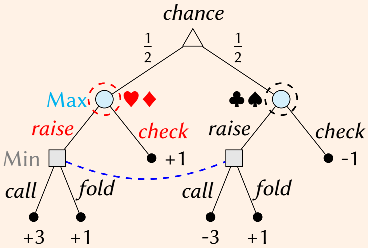
There are “simple” classes of imperfect information games solvable in PTIME

Our contributions in this work

- ♠ New PTIME solvable class of imperfect information games by transformation to equivalent simpler games
- ♠ Generalize transformation technique to broader class of games

Games in Extensive form

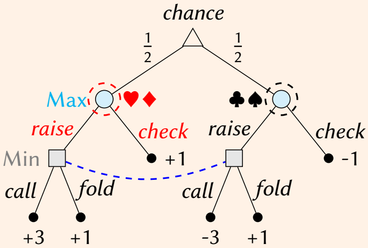
Player nodes : Max ◯ and Min ☐
Random node : Chance △



Games in Extensive form

Player nodes : Max ◯ and Min ◻
Random node : Chance △

Toy Poker : A random suit from ♥, ♣, ♦, ♠ is privately dealt to Max

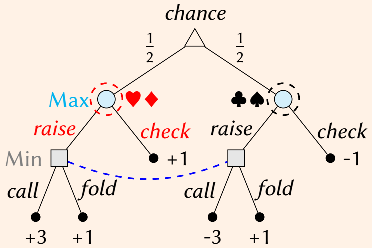


Games in Extensive form

Player nodes : Max ○ and Min □

Random node : Chance △

Toy Poker : A random suit from ♥, ♣, ♦, ♠ is privately dealt to Max

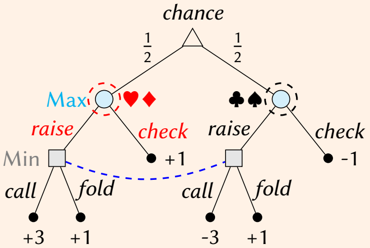


Actions : $A_{\text{Max}} = \{\textit{raise}, \textit{check}, \textit{raise}, \textit{check}\}$, $A_{\text{Min}} = \{\textit{call}, \textit{fold}\}$

Games in Extensive form

Player nodes : Max ○ and Min □
Random node : Chance △

Toy Poker : A random suit from ♥, ♣, ♦, ♠ is privately dealt to Max

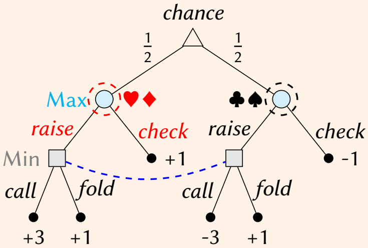


Actions : $A_{Max} = \{raise, check, raise, check\}, A_{Min} = \{call, fold\}$
Information sets : $\mathcal{I}_{Max} = \{--, --\}, \mathcal{I}_{Min} = \{--\}$

Games in Extensive form

Player nodes : Max ○ and Min □
Random node : Chance △

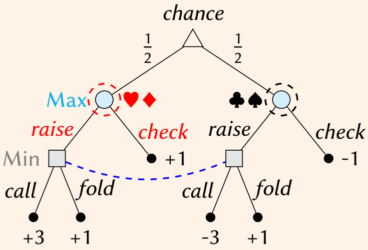
Toy Poker : A random suit from ♥, ♣, ♦, ♠ is privately dealt to Max



Actions : $A_{\text{Max}} = \{raise, check, raise, check\}$, $A_{\text{Min}} = \{call, fold\}$
Information sets : $\mathcal{I}_{\text{Max}} = \{--, --\}$, $\mathcal{I}_{\text{Min}} = \{--\}$
Zero-sum : Min pays Max at leaves ●

Strategies

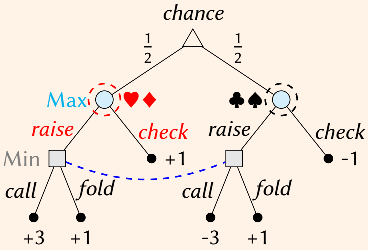
Strategies



Behavioral Strategies : Choose actions randomly at information set.

$$\mathcal{I} \mapsto \Delta(A)$$

Strategies



Behavioral Strategies : Choose actions randomly at information set.
$\mathcal{I} \mapsto \Delta(A)$

σ: Max raises and checks with probability $\frac{1}{2}$ at ♥♦ and always raises at ♣♠

τ: Min always folds

Expected Payoff : $E(\sigma, \tau)$

Optimal Solution

Optimal Solution

Maxmin value

$$\mathbf{maxmin} = \max_{\sigma} \min_{\tau} E(\sigma, \tau)$$

Optimal Solution

Maxmin value

$$\textbf{maxmin} = \max_{\sigma} \min_{\tau} E(\sigma, \tau)$$

Decision Problem

Optimal Solution

Maxmin value

$$\textbf{maxmin} = \max_{\sigma} \min_{\tau} E(\sigma, \tau)$$

Decision Problem

Two-player : Given a two-player game and threshold λ, is the maxmin value over behavioral strategies at least λ?

One-player : Given a one-player game and thereshold λ, is the maximum value over behavioral strategies at least λ?

Optimal Solution

Maxmin value

$$\textbf{maxmin} = \max_{\sigma} \min_{\tau} E(\sigma, \tau)$$

Decision Problem

Two-player : Given a two-player game and threshold λ, is the maxmin value over behavioral strategies at least λ?

One-player : Given a one-player game and thereshold λ, is the maximum value over behavioral strategies at least λ?

Complexity depends on player's recall.

Perfect recall

Perfect recall

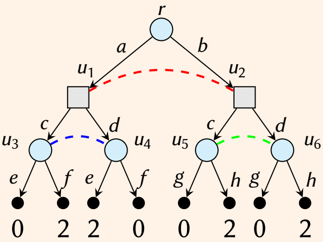
Player remembers his own past history of actions

$$inf(u) = inf(v) \implies hist_{\mathsf{Max}}(u) = hist_{\mathsf{Max}}(v)$$

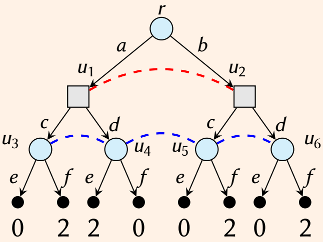
Perfect recall

Player remembers his own past history of actions

$$inf(u) = inf(v) \implies hist_{\text{Max}}(u) = hist_{\text{Max}}(v)$$



Perfect recall



Imperfect recall

No information set across distinct action sub-trees

Action-loss recall

Action-loss recall

Loss of perfect recall → distinct histories split at some past info set with distinct actions

Action-loss recall

Loss of perfect recall → distinct histories split at some past info set with distinct actions

$$inf(u) = inf(v) \implies$$

$$hist_{\mathsf{Max}}(u) = hist_{\mathsf{Max}}(v)$$

or

$hist_{\mathsf{Max}}(u) = sa_1s_1$, $hist_{\mathsf{Max}}(v) = sa_2s_2$ where a_1, a_2 action at same info set

Action-loss recall

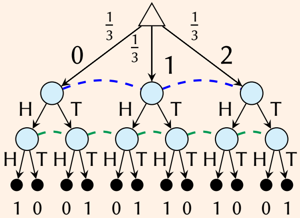
Loss of perfect recall → distinct histories split at some past info set with distinct actions

$$inf(u) = inf(v) \implies$$

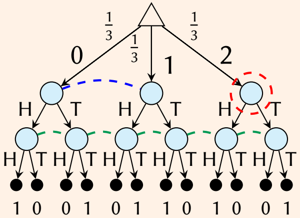
$hist_{Max}(u) = hist_{Max}(v)$

or

$hist_{Max}(u) = sa_1 s_1$, $hist_{Max}(v) = sa_2 s_2$ where a_1, a_2 action at same info set



A-loss recall



Not A-loss recall

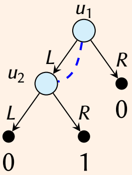
No information set across sub-trees rooted at nodes with distinct information but same history

Absentmindedness

Absentmindedness

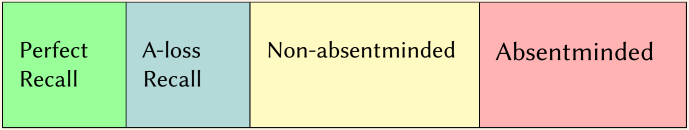
Forgets if the same decision point was seen before

$$\exists u, v,\ inf(u) = inf(v) \text{ and } hist_{\text{Max}}(u) <_{prefix} hist_{\text{Max}}(v)$$



Absentminded

Previous Complexity Picture



Previous Complexity Picture

Increasing complexity →

Perfect Recall	A-loss Recall	Non-absentminded	Absentminded
PTIME	PTIME	NP-complete	$\exists\mathbb{R}$-complete
[K,M'92]	[K,K'95]	[K,M'92]	[G,P,S'20]

Complexity of one-player games

Previous Complexity Picture

Increasing complexity ⟶

Perfect Recall	A-loss Recall	Non-absentminded	Absentminded
PTIME [K,M'92]	PTIME [K,K'95]	NP-complete [K,M'92]	$\exists\mathbb{R}$-complete [G,P,S'20]

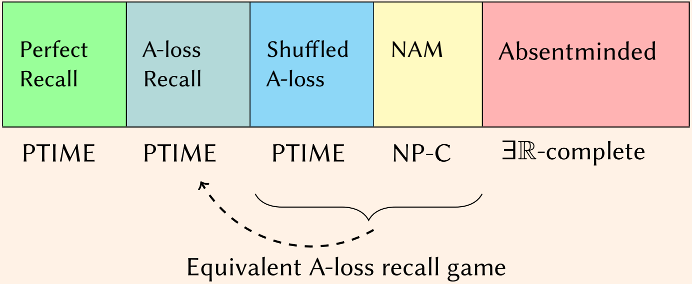
Complexity of one-player games

Our work : Finer complexity picture for non-absentminded games

Our contribution

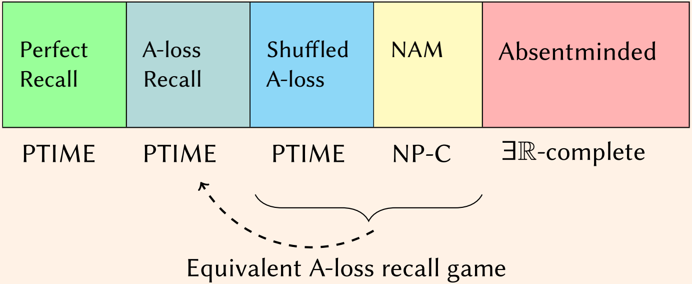
Perfect Recall	A-loss Recall	Shuffled A-loss	NAM	Absentminded
PTIME	PTIME	PTIME	NP-C	$\exists\mathbb{R}$-complete

Our contribution



Every non-absentminded game can be transformed into equivalent A-loss recall game.

Our contribution



Every non-absentminded game can be transformed into equivalent A-loss recall game.

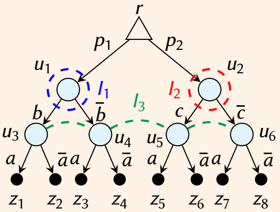
Caveat : Exponential blow-up in size

Core idea

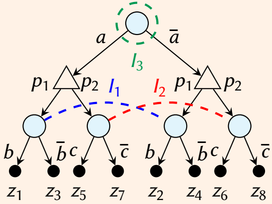
Assigining variables to actions for a behavioral strategy gives symbolic payoff polynomial

Games are equivalent if symbolic polynomials are same

Shuffled A-loss Recall



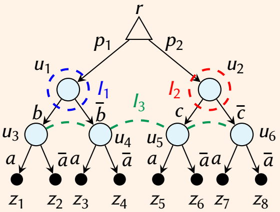
No A-loss recall



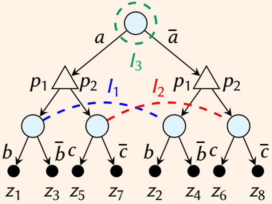
A-loss recall

Actions in histories are permuted

Shuffled A-loss Recall



No A-loss recall



A-loss recall

Actions in histories are permuted

Theorem

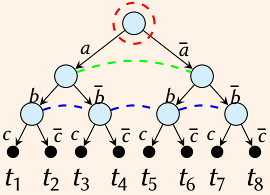
Shuffled A-loss recall can be detected and computed in PTIME

A-loss recall span

A-loss recall span

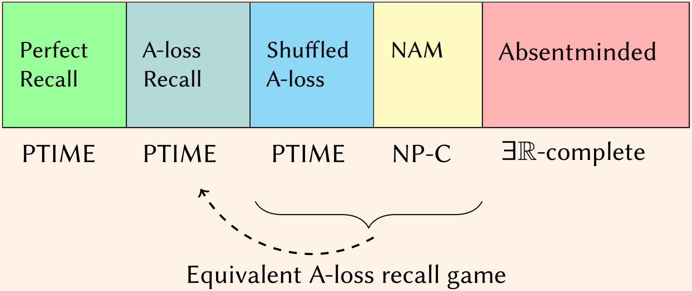
Leaf polynomials of input game are linear combinations of leaf polynomials of transformed game

Set of polynomials $\{x_a, 1 - x_a\} \times \{x_b, 1 - x_b\} \times \{x_c, 1 - x_c\}$ forms basis of vector space of all multilinear polynomials over $\{x_a, x_b, x_c\}$



Add suitable payoffs t_i to construct equivalent game

Summary



Also extends to two-player games where Max has perfect recall and Min is non-absentminded

Future directions

- Implications on practice
- Complexity of computing minimal A-loss recall Span

Thank You