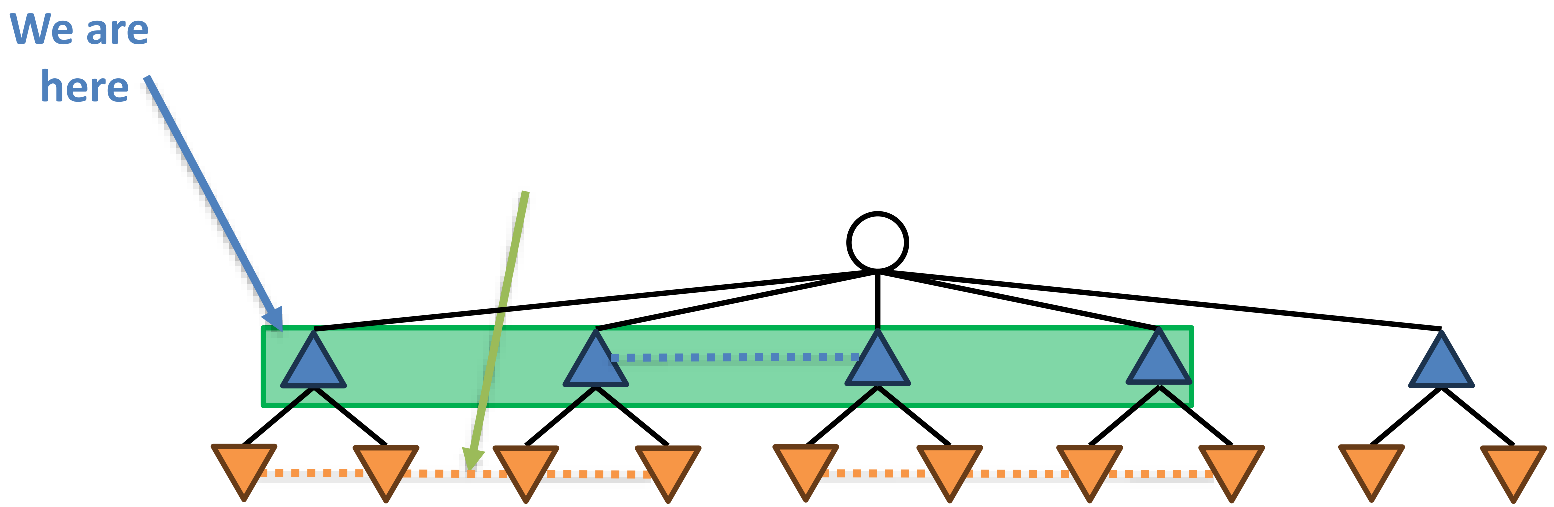
Knowledge-Limited Subgame Solving

Safe subgame resolving

Safe subgame solving is based on the common-knowledge subgame

Definition:

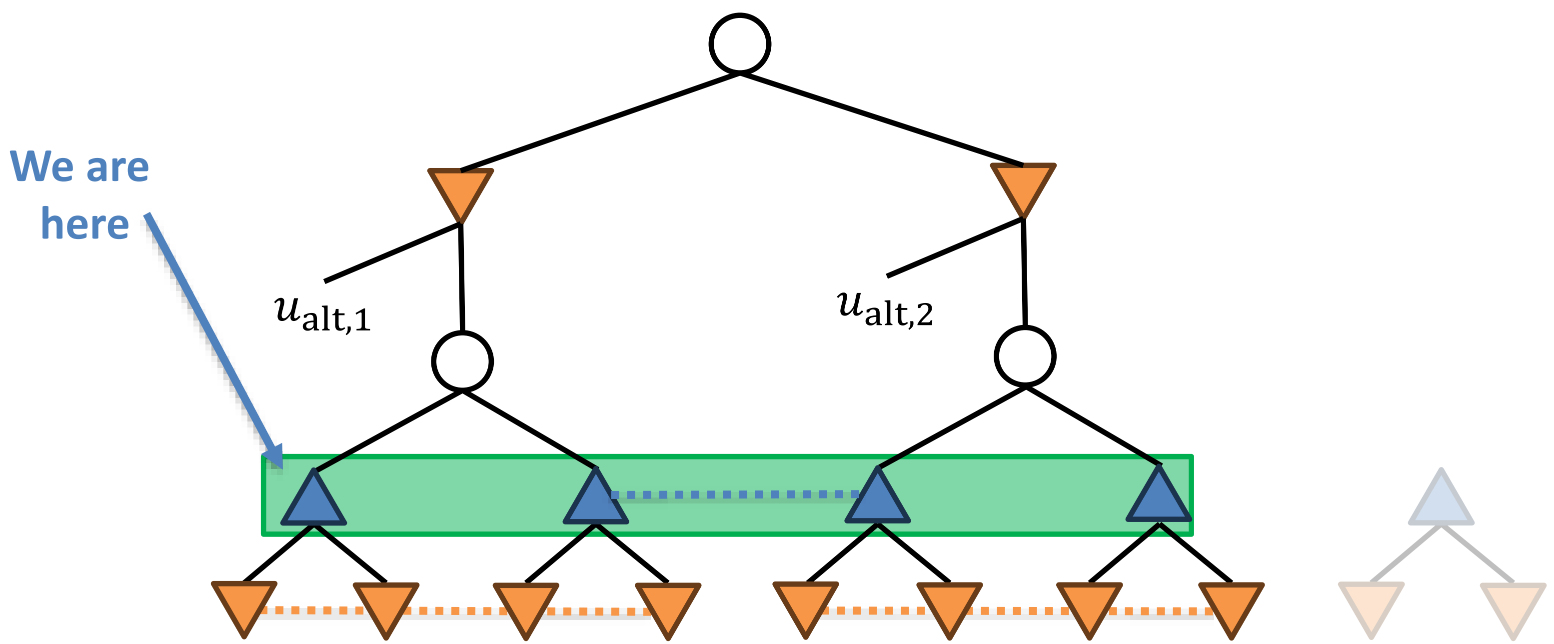
- Two nodes in the same layer of the game tree are *connected* if there is an infoset connecting some descendant of the first node to some descendant of the second node
- The **common-knowledge subgame** at a node h consists of all nodes *recursively* connected to h, and all their descendants.



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

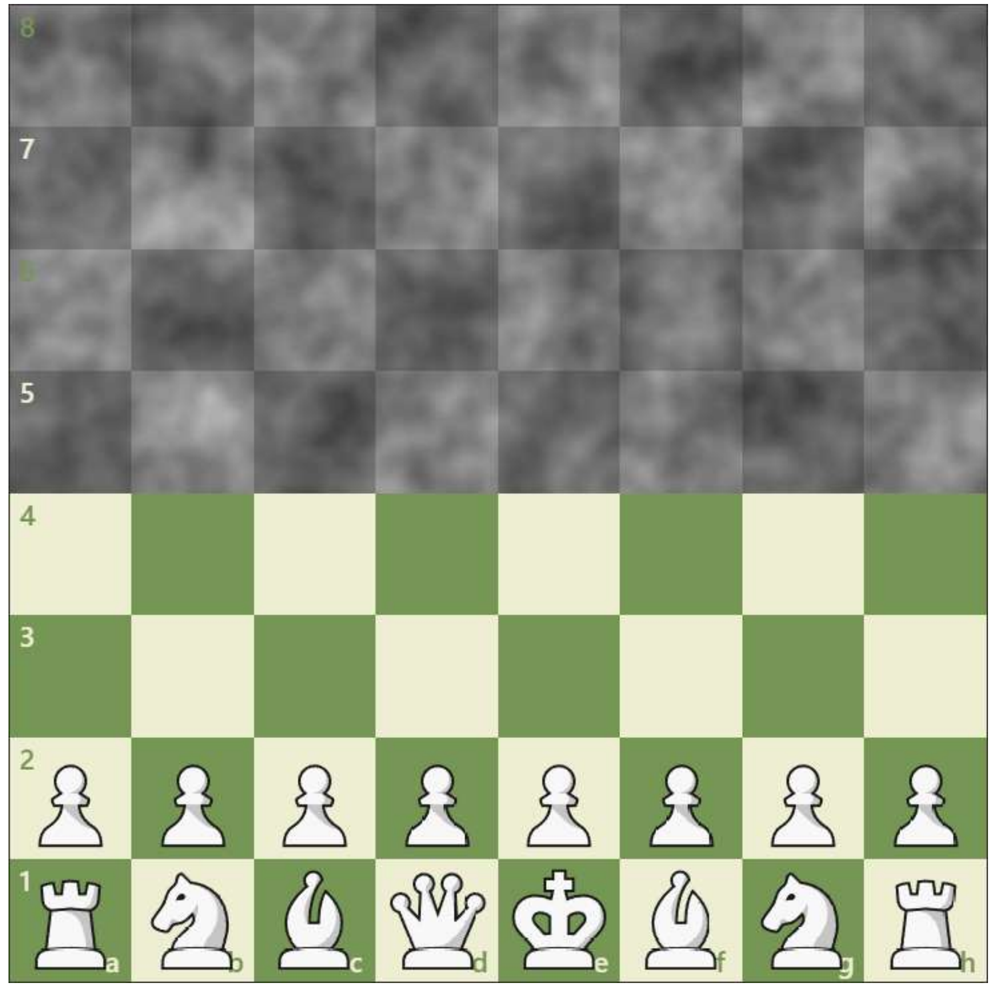
Safe subgame resolving

Resolve gadget game:



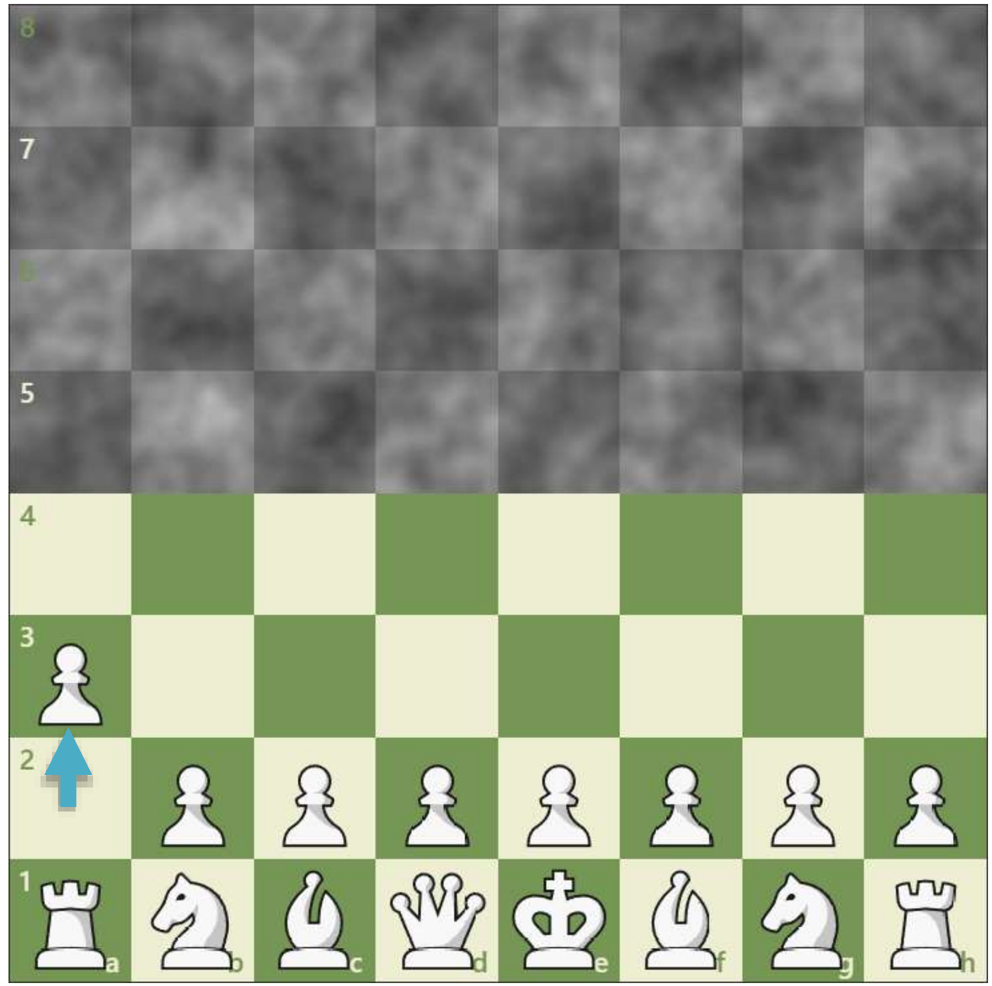
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



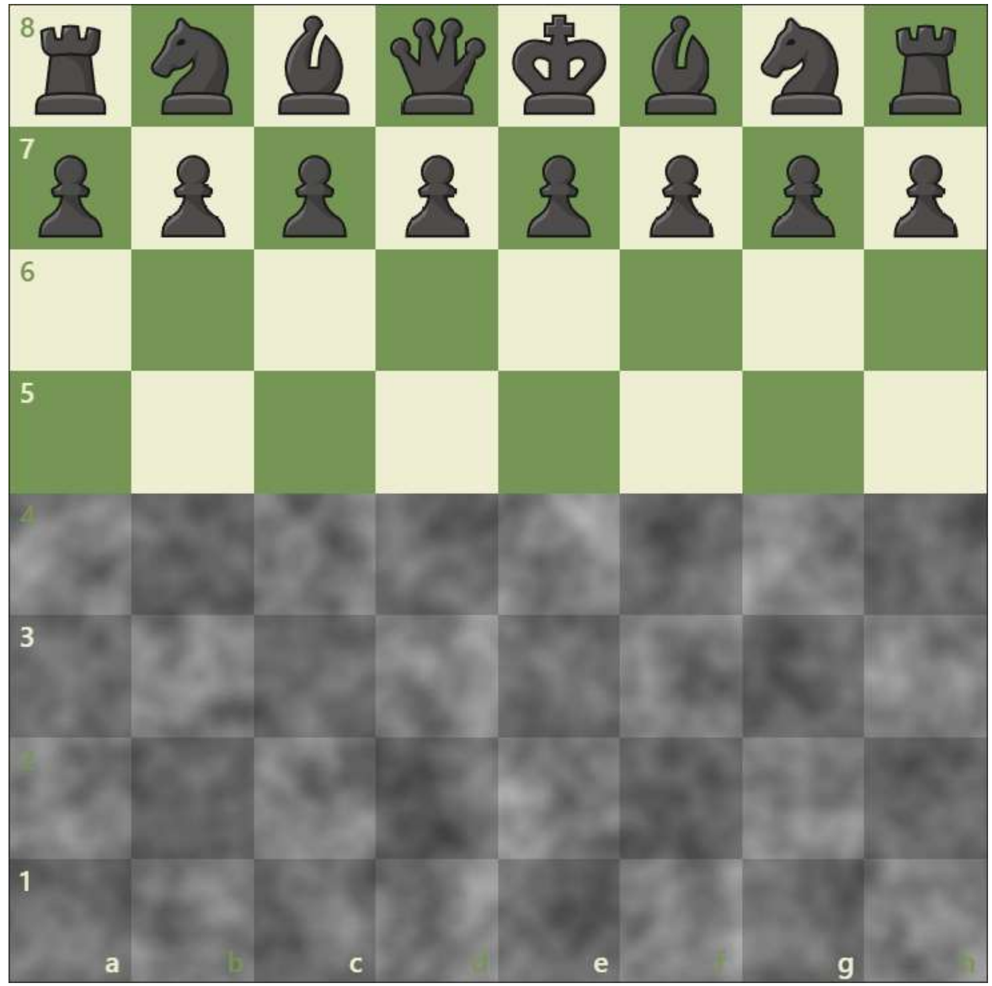
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



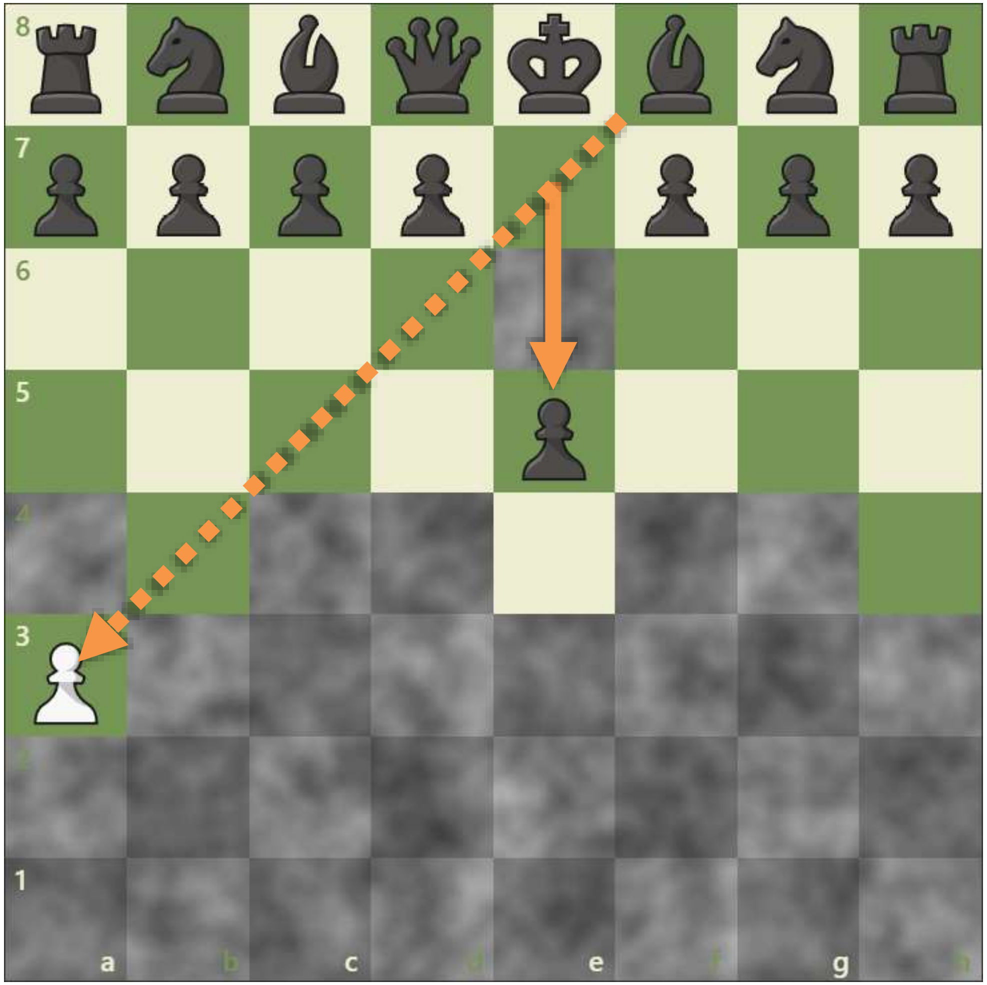
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



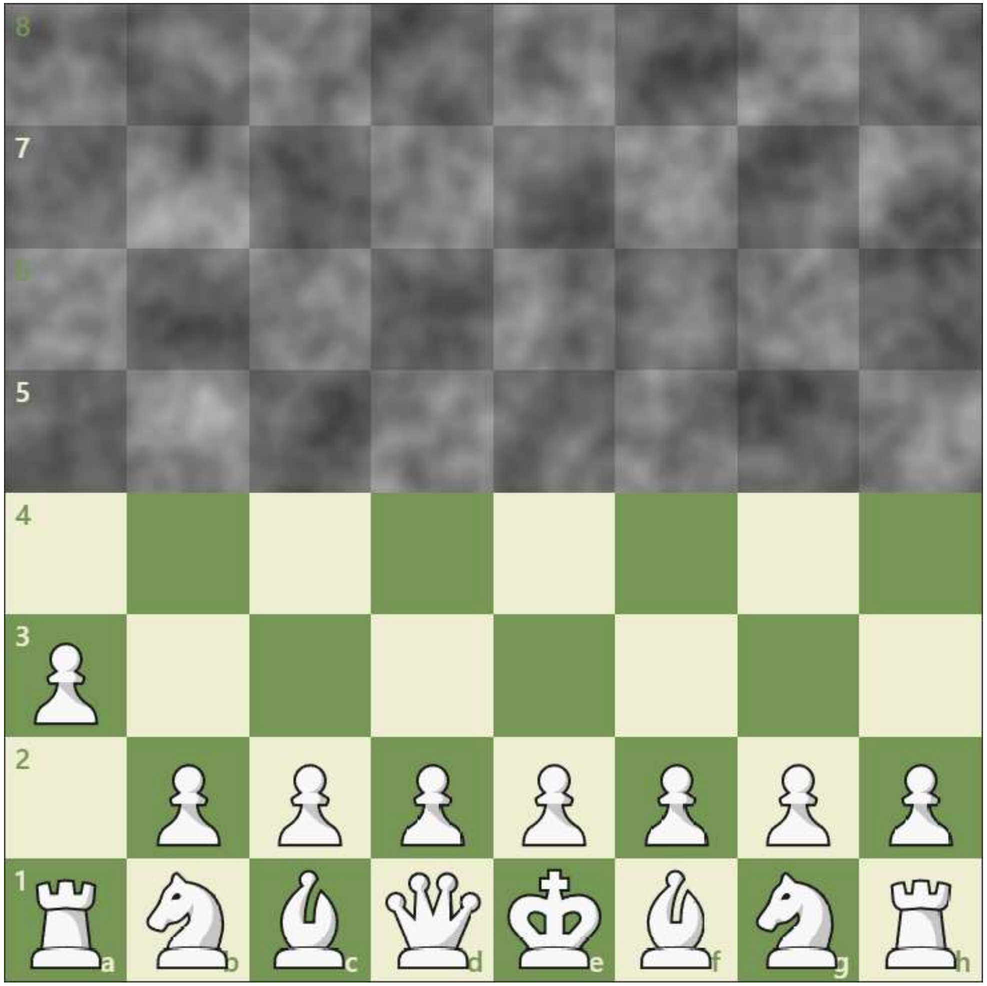
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



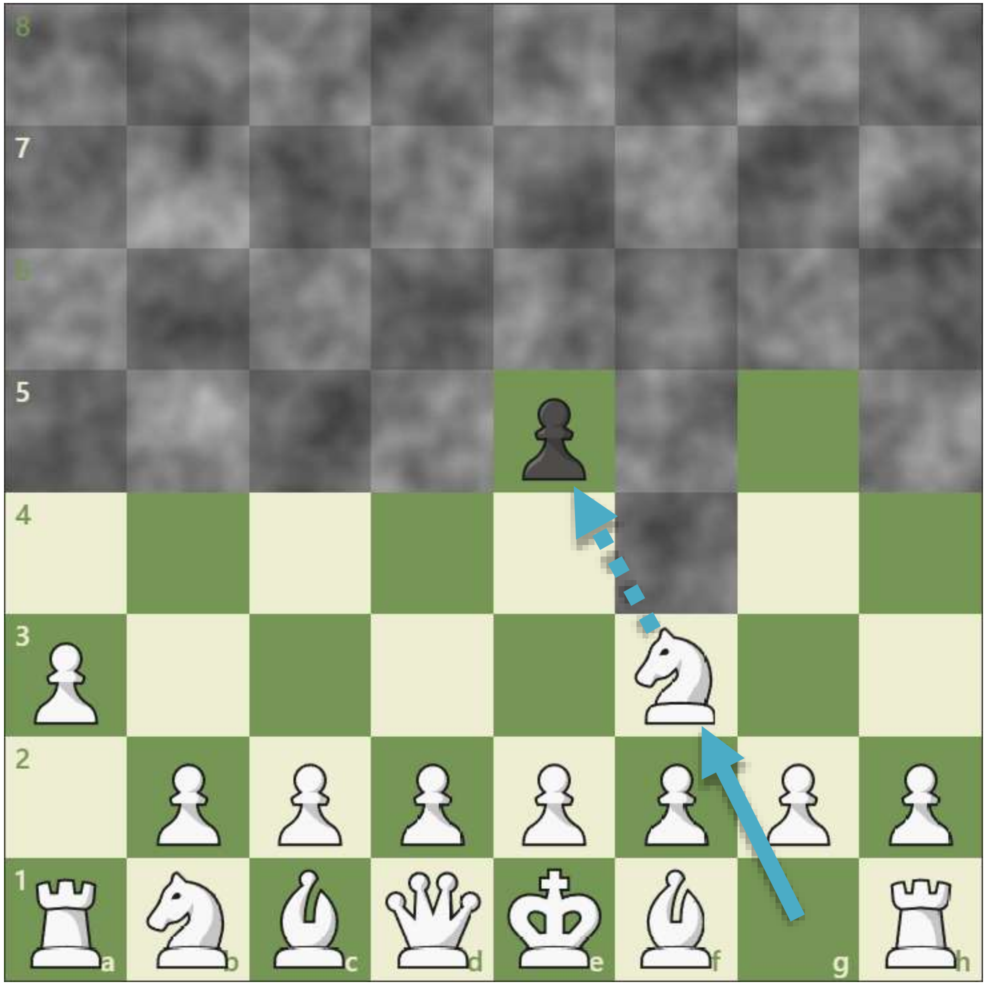
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



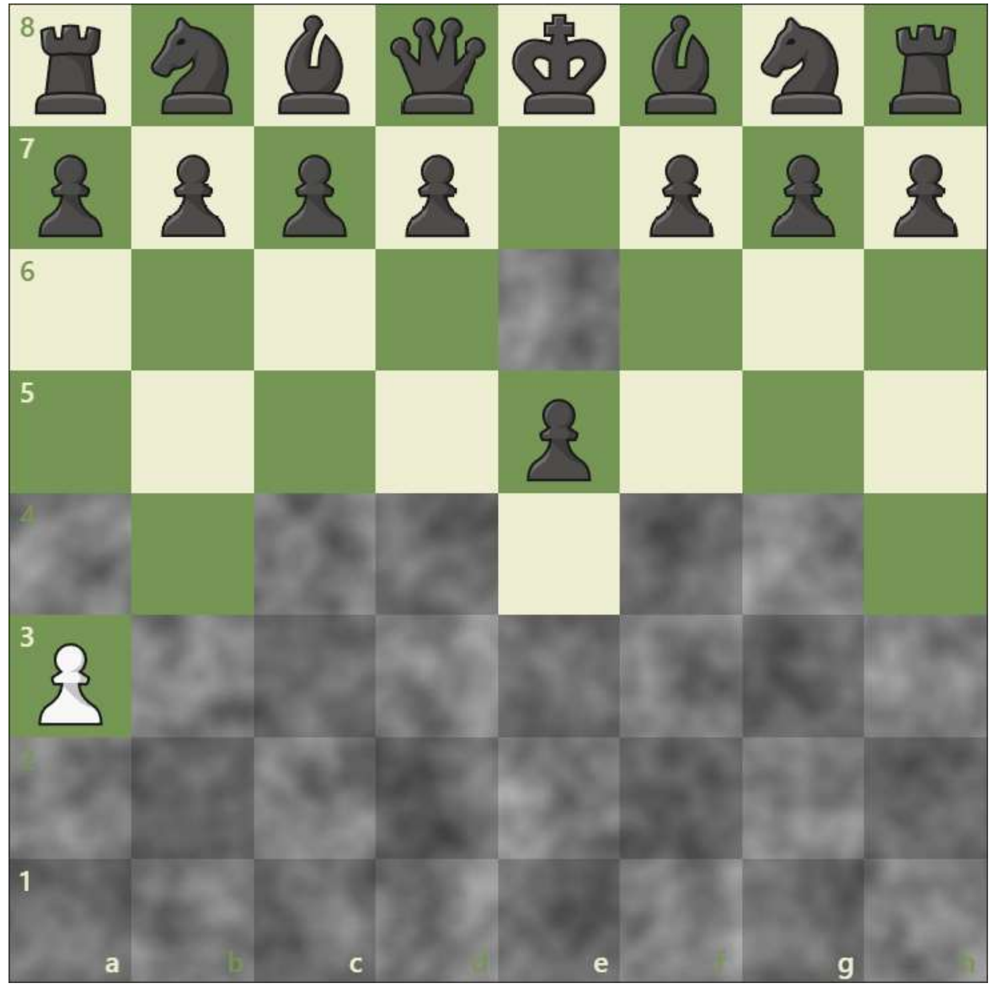
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



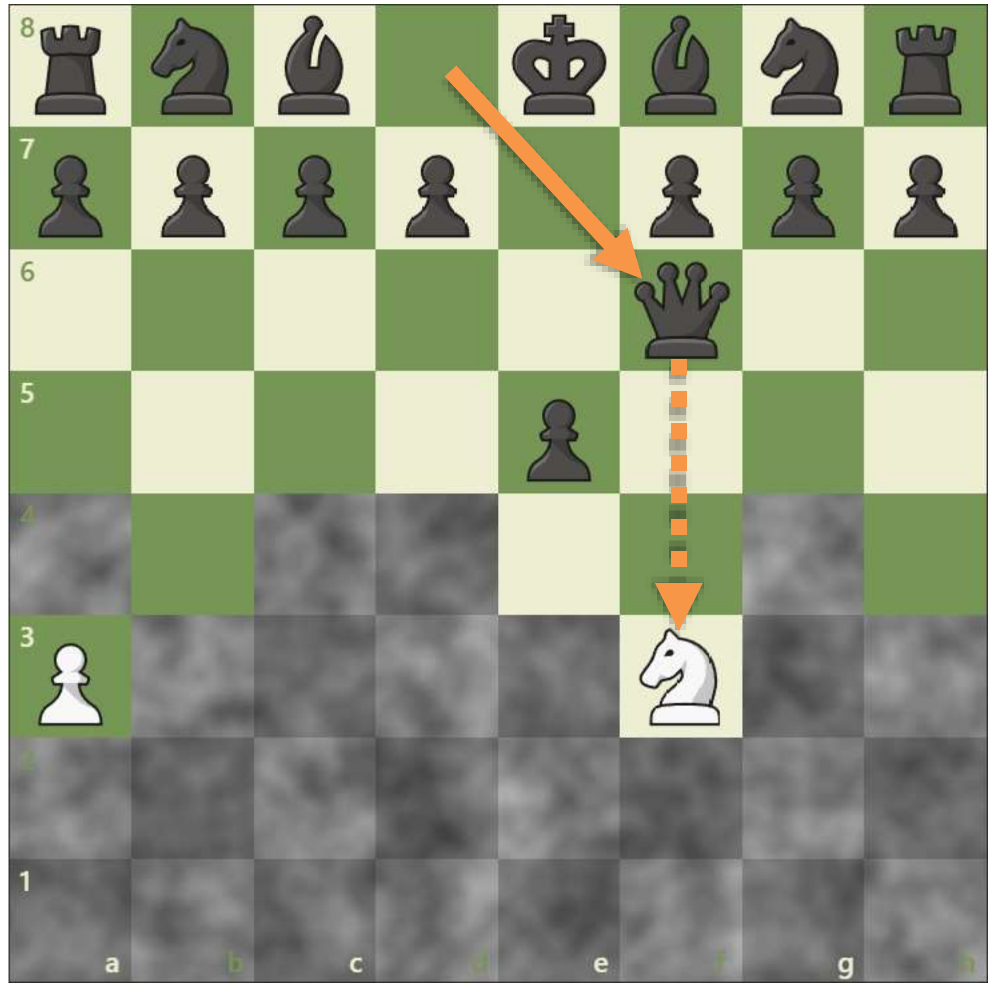
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

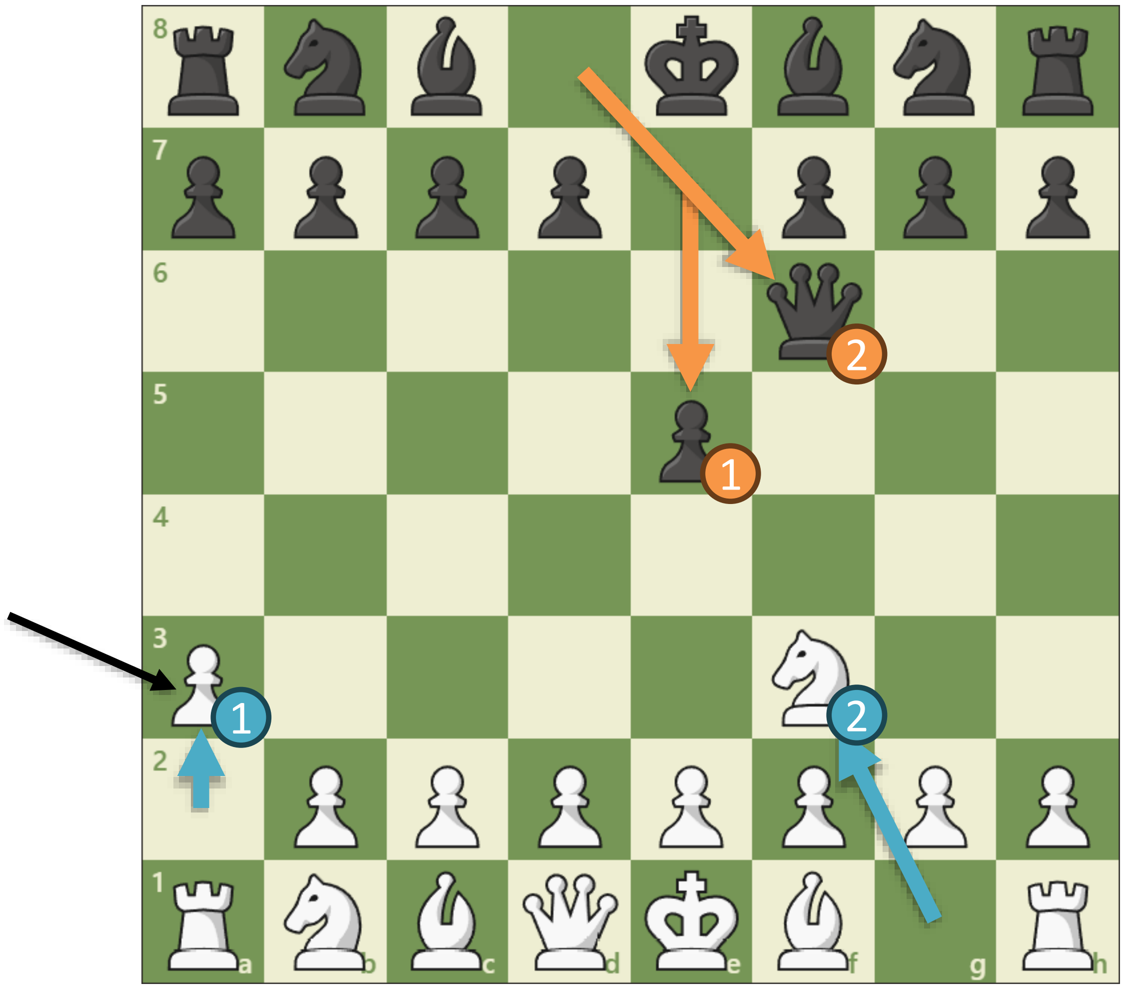
Dark chess, a.k.a. Fog of War Chess



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess

There is a pawn on a3.



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess

Black knows that
White knows that
Black knows that
White knows that
there is a pawn on a3.

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess

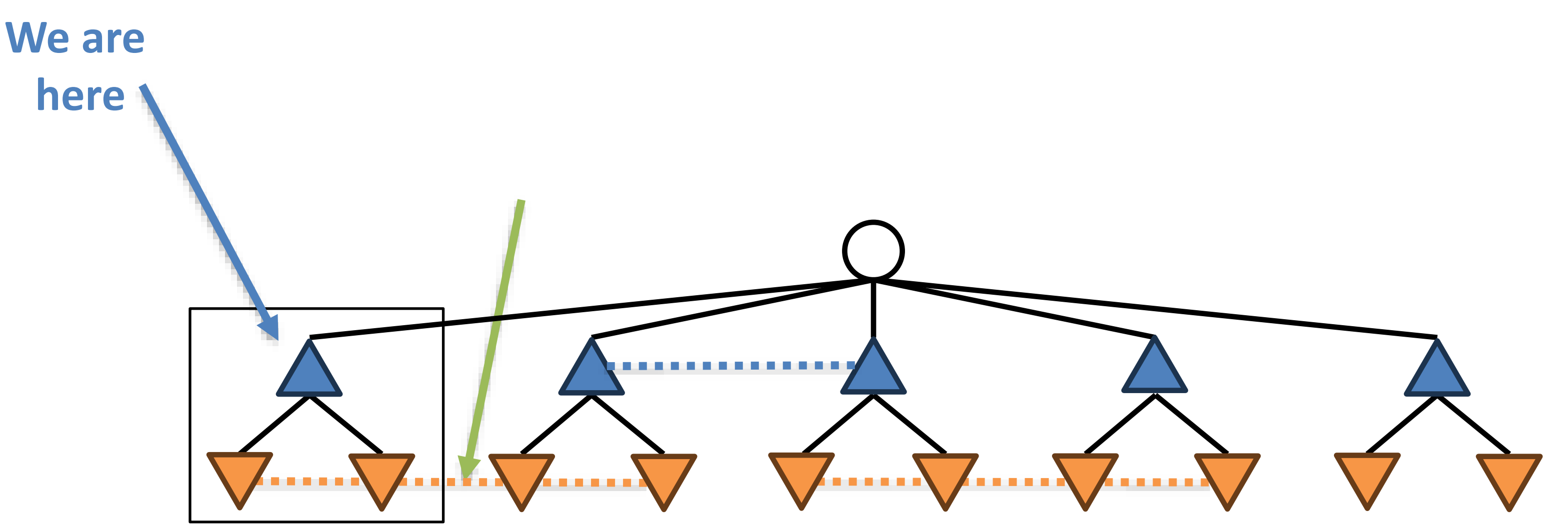
Does White know that
Black knows that
White knows that
Black knows that
White knows that
there is a pawn on a3?

No! White didn't see the queen

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Subgame solving in imperfect-information games

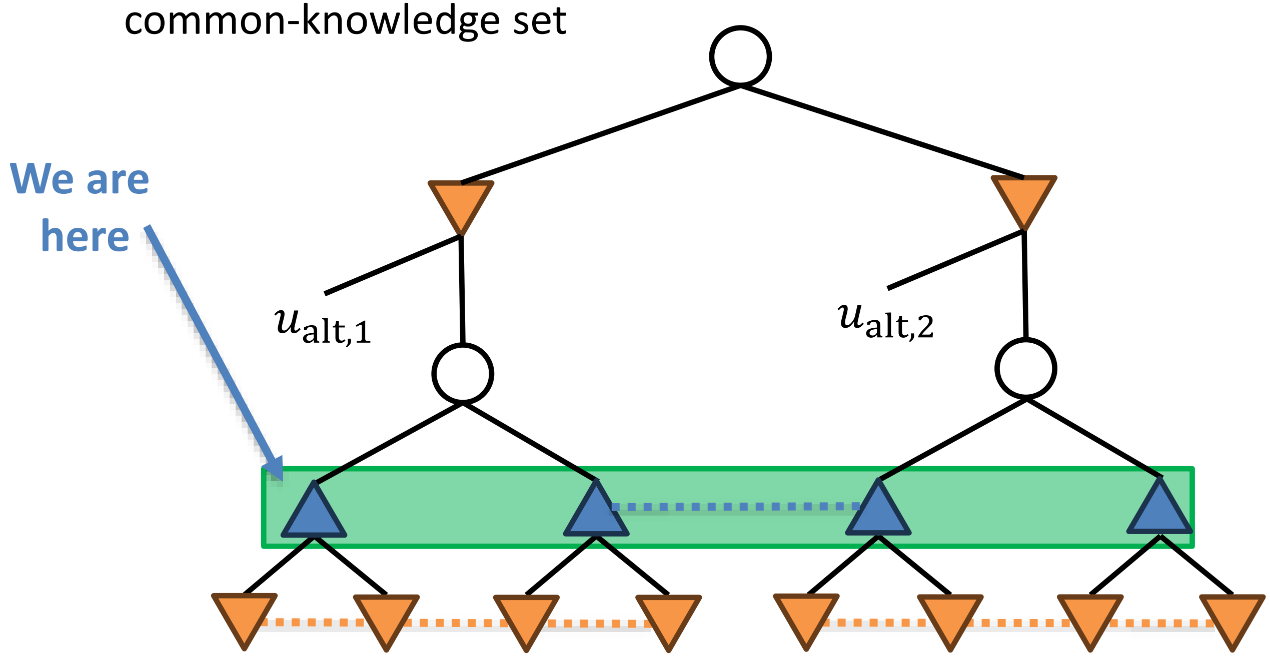
We cannot solve the subgame in isolation, because the solution may depend on the remainder of the game



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

So, what is the problem?

- Common-knowledge sets can be very large!
  - Heads-up Texas hold'em: $|C| < 2 \times 10^6$
    - Manageable in real time
    - Practical tricks [Johanson et al *IJCAI*-11] mean that, effectively, $C \approx 10^3$
  - Dark chess: Common for C to be too large to store in memory, much less work with in real time
    - Not even obvious how to determine whether two nodes are in the same common-knowledge set



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess

Does White know that
Black knows that
White knows that
Black knows that
White knows that
there is a pawn on a3?

No! White didn't see the queen

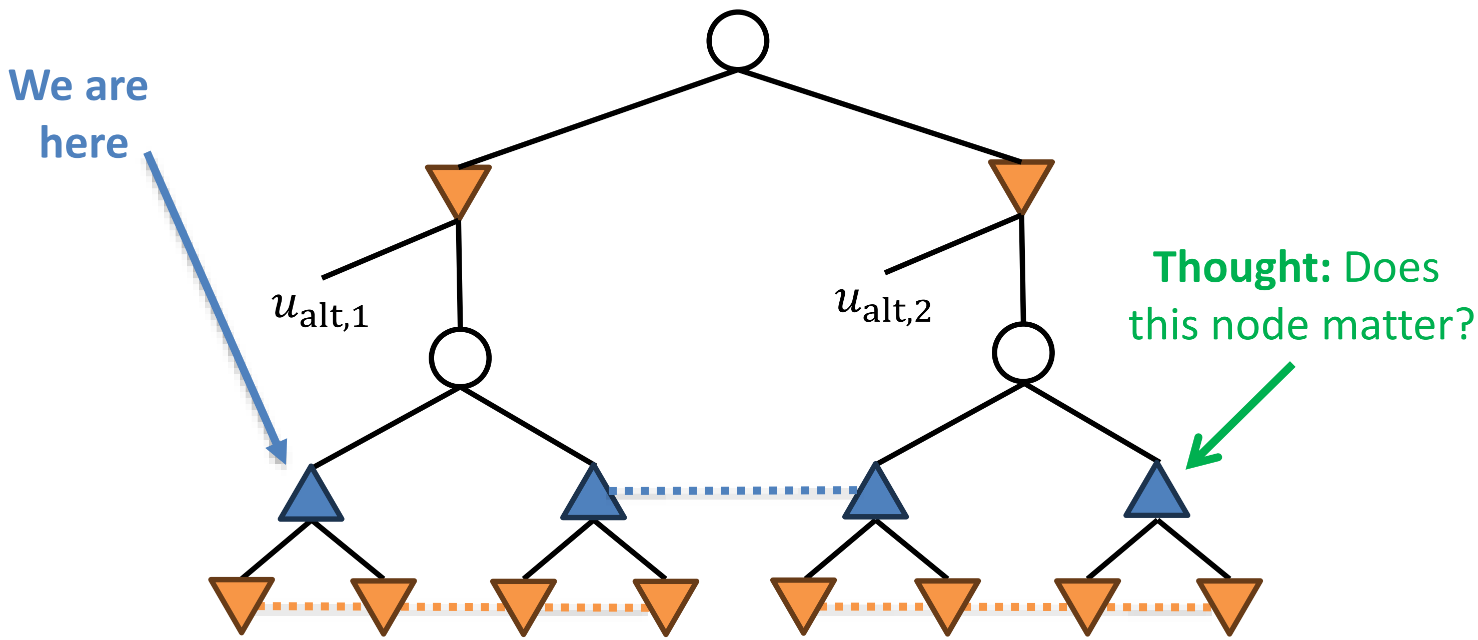
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess, a.k.a. Fog of War Chess

This game state is in the same common-knowledge set as the previous one!

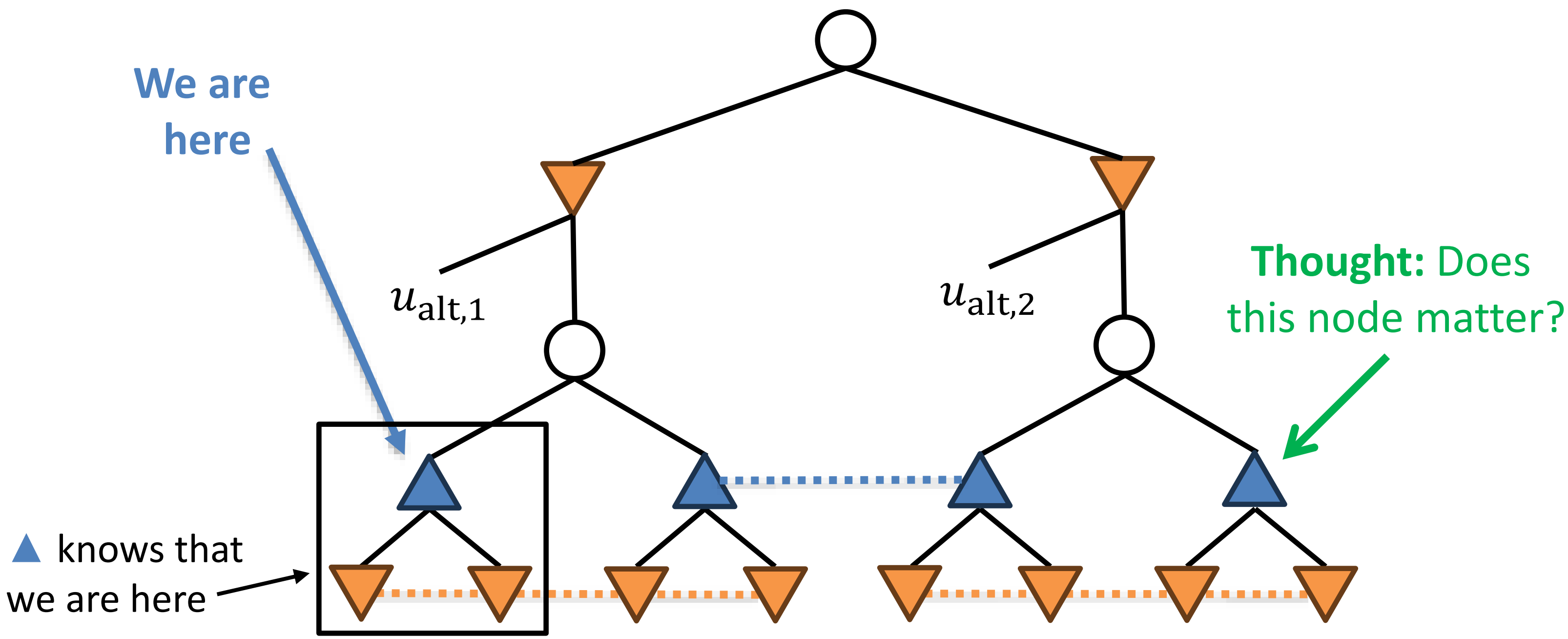
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Knowledge-limited subgame solving



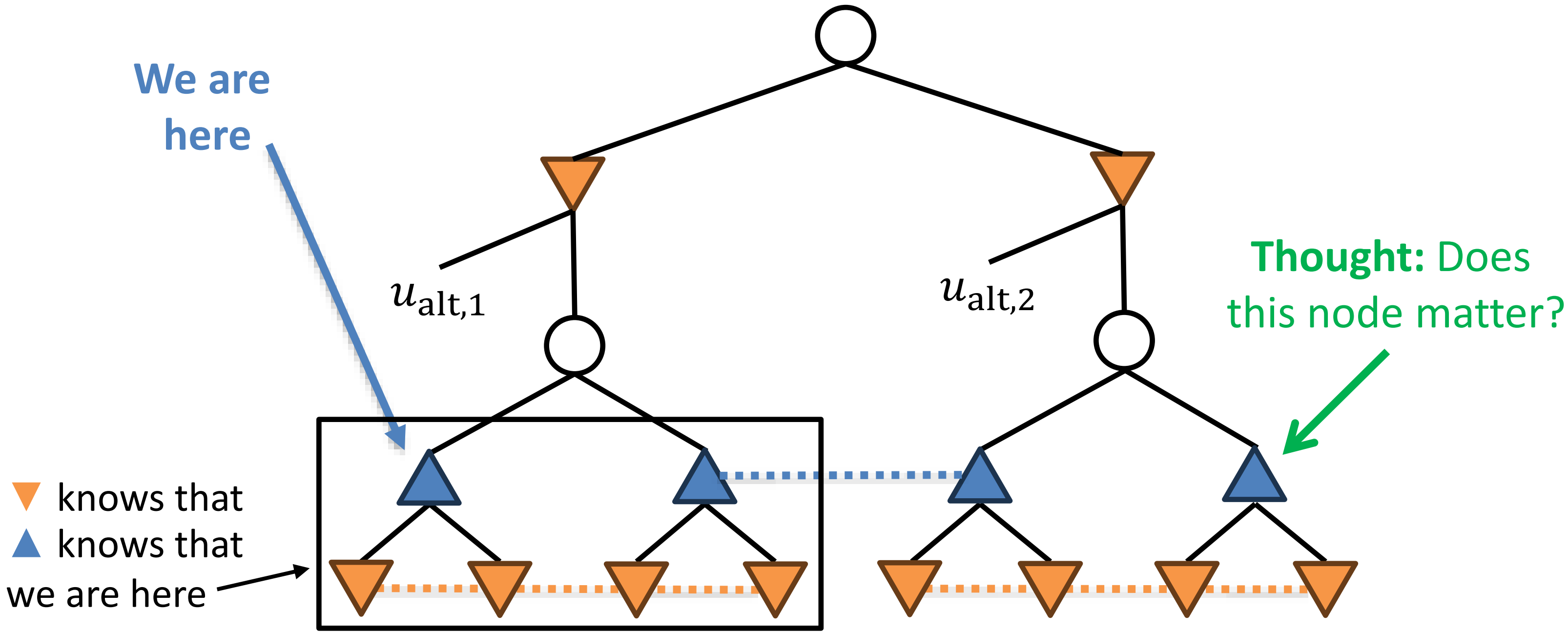
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Knowledge-limited subgame solving



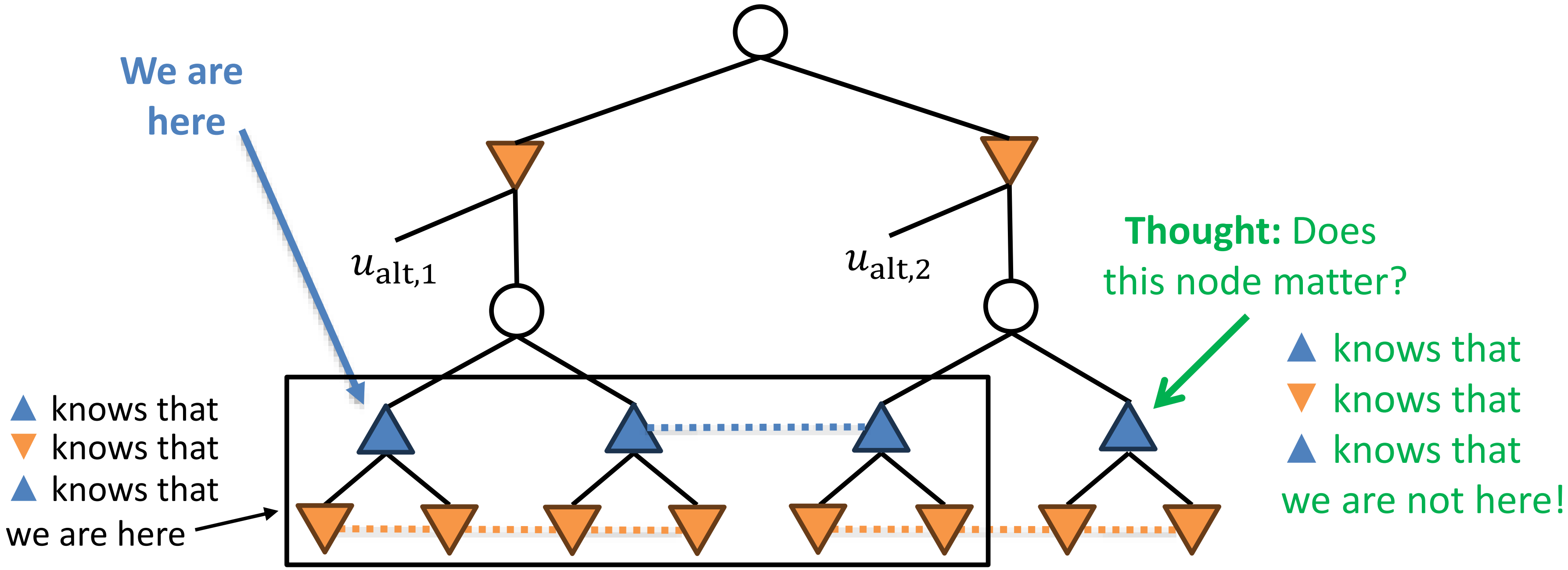
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Knowledge-limited subgame solving



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Knowledge-limited subgame solving

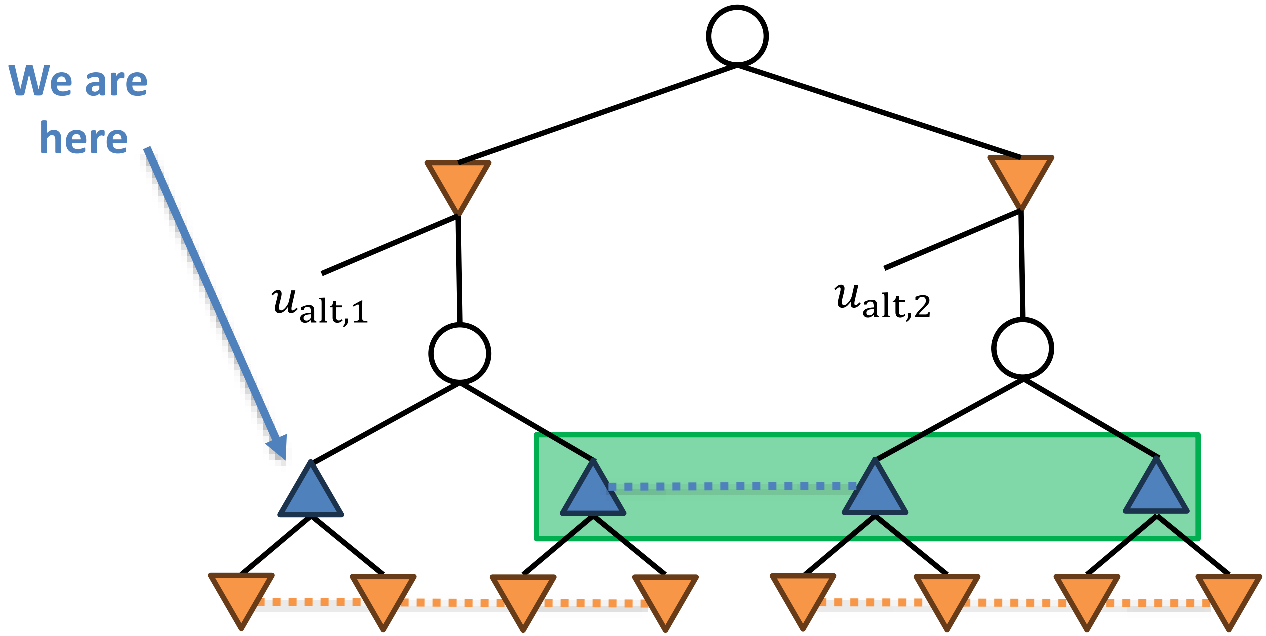


Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Knowledge-limited subgame solving

Idea: Assume that we will not deviate from the blueprint at nodes no longer reachable

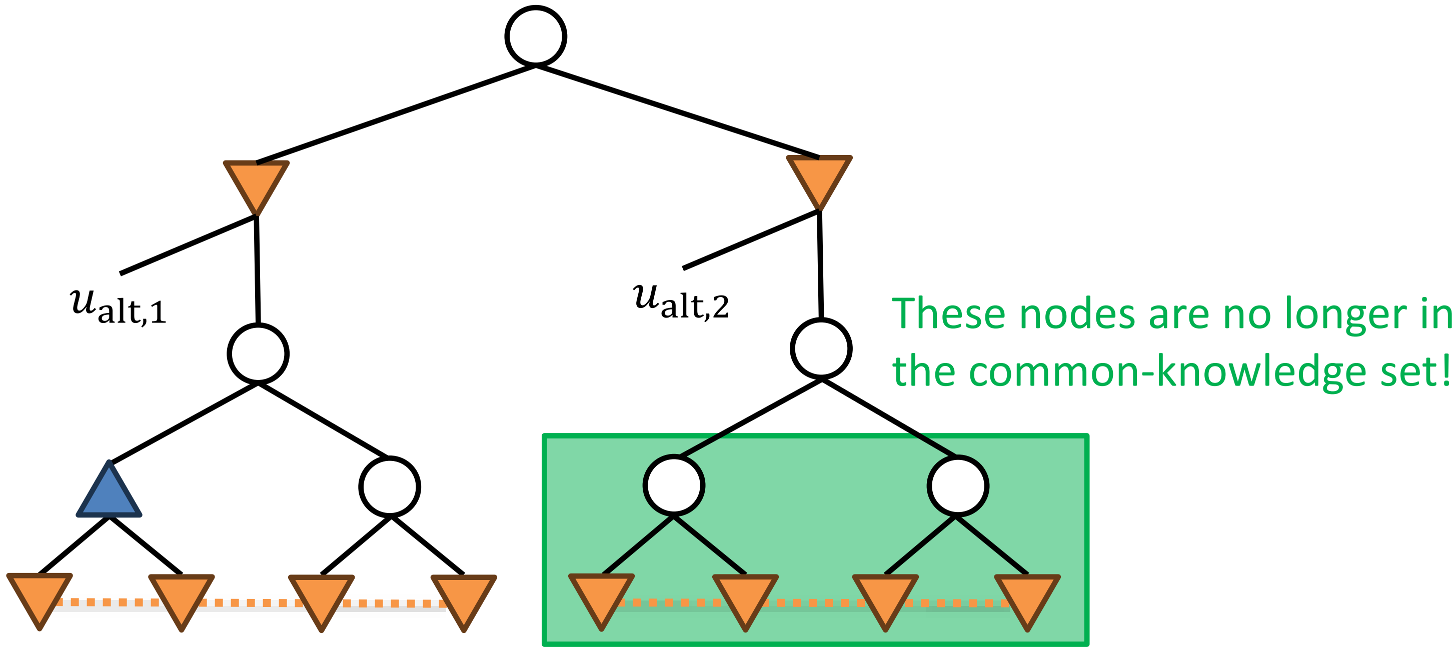
- They become chance nodes with fixed probabilities



Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

1-Knowledge-limited subgame solving

Now let us see what happens if we run subgame solving as usual



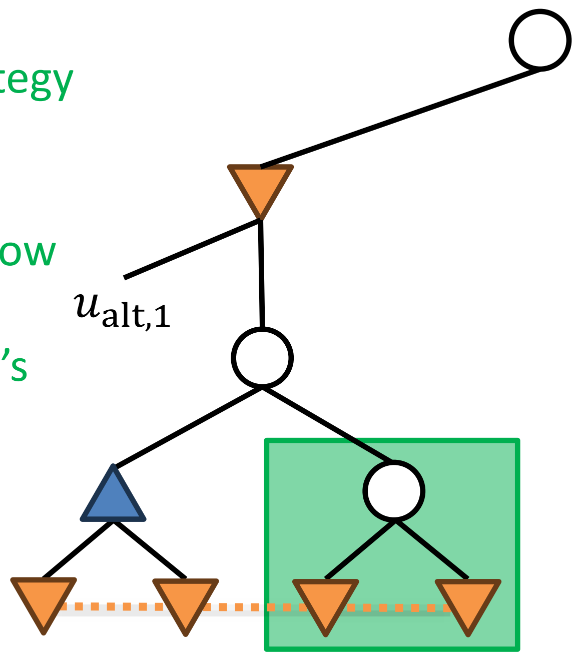
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

1-Knowledge-limited subgame solving

Now let us see what happens if we run subgame solving as usual

These nodes only depend on ▼'s strategy

They can be represented in the row of the payoff matrix corresponding to ▲'s root sequence



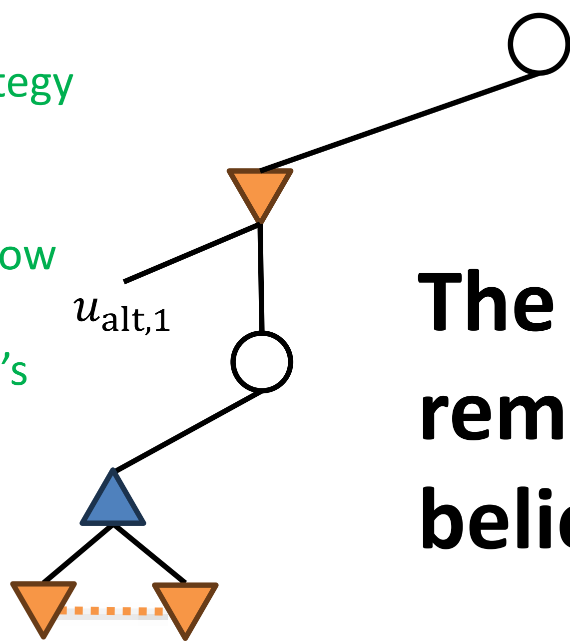
Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

1-Knowledge-limited subgame solving

Now let us see what happens if we run subgame solving as usual

These nodes only depend on ▼'s strategy

They can be represented in the row of the payoff matrix corresponding to ▲'s root sequence



The only nodes that remain are those that we believe are reachable!

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Is it safe in theory?

No. Easy counterexample:

- N copies of matching pennies. Chance chooses the copy. Max knows which copy we are playing, but Min does not
- **Blueprint:** Max plays Heads w.p. $\frac{1}{2} + \frac{2}{N}$ in all copies
- **KLSS:** With all other infosets fixed, Max switching to "always play tails" in one infoset results in a more balanced strategy
- Thus, after KLSS, Max always plays tails!
- **Intuition:** KLSS will *overcorrect for systematic errors* in the blueprint

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Is it safe in practice?

Yes!

- Even when the blueprint has bad systematic errors (the blueprints in these experiments were ε-noisy Nash equilibria for $\varepsilon = 0.25$)

game	exploitability blueprint	after 1-KLSS	ratio
2x2 Abrupt Dark Hex	.0683	.0625	1.093
4-card Goofspiel, random order	.171	.077	2.2
4-card Goofspiel, increasing order	.17	.0	∞
Kuhn poker	.0124	.0015	8.3
Kuhn poker (ε-bet)	.0035	.0	∞
3-rank limit Leduc poker	.0207	.0191	1.087
3-rank limit Leduc poker (ε-fold)	.0065	.0057	1.087
3-rank limit Leduc poker (ε-bet)	.0097	.0096	1.011
Liar's Dice, 5-sided die	.181	.125	1.45
100-Matching pennies	.0013	.0098	0.13

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess

- Large game
  - Game tree is almost identical to that of regular chess
- Moderate-sized information sets
  - Size $\approx 10^5$ to 10^6 is common, but $> 10^7$ is very rare
- Unmanageable common-knowledge sets
  - Common-knowledge sets likely have size $> 10^{12}$
  - Enumeration is impractical in real time
- **Our techniques allow the creation of a strong agent!**

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess

Advantages over naïve techniques:

- Our bot can bluff and exploit the opponent’s lack of knowledge.
- Our bot can mix accurately, which is important in many situations.
  - A bot that does not mix at all in Dark Chess is easily exploitable!

Weaknesses of the bot:

- Due to how the bot handles information and chooses what nodes to expand, it assumes the opponent knows more than they actually do.
- “Iterative deepening” subgame search does not prune very well due to the randomness necessary to play dark chess well, so subgame solves are relatively low-depth and will miss tactics.

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Dark chess

Performance:

- Comfortably better than me
  - I am ≈1700 on chess.com
- Lost to the world #1 human (≈2400) 9-1
- Strong opening and middlegame play; weak endgame play (plays too conservatively)

Zhang and Sandholm, "Subgame solving without common knowledge", *NeurIPS* 2021

Safe KLSS

KLSS is unsafe. **Why?**

conditional best response value to x at J

conditional best response value to x_0 at J

$$\text{Margin}_J(x) = \min_{y_J} u_J(x, y_J) - \min_{y'_J} u_J(x_0, y'_J)$$

y_J, y'_J: strategy for ▼ following J
u_S: value conditional on reaching set S
x: ▲'s resolved strategy in subgame
x_0: ▲'s blueprint strategy
J: an infoset for ▼

Recall: Every strategy with positive margins is safe

Liu, Fu, Fu, Yang, "Opponent-Limited Online Search for Imperfect Information Games", *ICML* 2023

Safe KLSS

KLSS is unsafe. **Why?**

conditional best response value to x at J

conditional best response value to x_0 at J

$$\text{Margin}_J(x) = \min_{y_J} \left\{ u_J(x, y_J) - \min_{y'_J} u_J(x_0, y'_J) \right\}$$

y_J, y'_J: strategy for ▼ following J
u_S: value conditional on reaching set S
x: ▲'s resolved strategy in subgame
x_0: ▲'s blueprint strategy
J: an infoset for ▼

Recall: Every strategy with positive margins is safe

Liu, Fu, Fu, Yang, "Opponent-Limited Online Search for Imperfect Information Games", *ICML* 2023

Safe KLSS

KLSS is unsafe. **Why?**

$$\text{Margin}_J(\boldsymbol{x}) = \min_{\boldsymbol{y}_J} \left\{ u_J(\boldsymbol{x}, \boldsymbol{y}_J) - u_J(\boldsymbol{x}_0, \boldsymbol{y}_J) + u_J(\boldsymbol{x}_0, \boldsymbol{y}_J) - \min_{\boldsymbol{y}'_J} u_J(\boldsymbol{x}_0, \boldsymbol{y}'_J) \right\}$$

$\boldsymbol{y}_J, \boldsymbol{y}'_J$: strategy for ▼ following J
u_S: value conditional on reaching set S
$\boldsymbol{x}$: ▲'s resolved strategy in subgame
$\boldsymbol{x}_0$: ▲'s blueprint strategy
J: an infoset for ▼

Recall: Every strategy with positive margins is safe

Liu, Fu, Fu, Yang, "Opponent-Limited Online Search for Imperfect Information Games", *ICML* 2023

Safe KLSS

KLSS is unsafe. **Why?**

Unsafe KLSS assumes only one of these terms can be nonzero at a time

$$\text{Margin}_J(x) = \min_{y_J}\left\{\sum_I \beta(I)\left(u_{I\cap J}(x_I, y_J) - u_{I\cap J}(x_{0;I}, y_J)\right) + u_J(x_0, y_J) - \min_{y_J'} u_J(x_0, y_J')\right\}$$

y_J, y_J': strategy for ▼ following J

u_S: value conditional on reaching set S

x: ▲'s resolved strategy in subgame

x_0: ▲'s blueprint strategy

J: an infoset for ▼

I: an infoset for ▲

$x_I, x_{0,I}$: ▲'s resolved/blueprint following I

$\beta(I)$: P1 reach probability of I according to blueprint

Recall: Every strategy with positive margins is safe

Liu, Fu, Fu, Yang, "Opponent-Limited Online Search for Imperfect Information Games", *ICML* 2023

Theorem: This is a safe way to do KLSS!

Safe KLSS

KLSS is unsafe. **Why?**

Unsafe KLSS assumes only one of these terms can be nonzero at a time

$$\text{Margin}_{I,J}(x) = \min_{y_J} \Big\{ \cancel{\beta(I)} \left(u_{I\cap J}(x_I, y_J) - u_{I\cap J}(x_{0;I}, y_J) \right)$$

"allows one infoset I to steal the whole gift"

Solution: Use gift-splitting, like in reach maxmargin!

$$+ u_J(x_0, y_J) - \min_{y_J'} u_J(x_0, y_J') \Big\}$$

difference between y_J's value at J against blueprint, and its best response value against blueprint

gift

y_J, y_J': strategy for ▼ following J

u_S: value conditional on reaching set S

x: ▲'s resolved strategy in subgame

x_0: ▲'s blueprint strategy

J: an infoset for ▼

I: an infoset for ▲

$x_I, x_{0,I}$: ▲'s resolved/blueprint following I

$\beta(I)$: P1 reach probability of I according to blueprint

Recall: Every strategy with positive margins is safe

Liu, Fu, Fu, Yang, "Opponent-Limited Online Search for Imperfect Information Games", *ICML* 2023

Safe KLSS

Game	Blueprint	Unsafe	Maxmargin	Resolving	1-KLSS	Safe-1-KLSS
Leduc(2,3,1)	0.150	0.195	0.109	**0.089**	0.190	0.120
Leduc(2,3,1)	0.150	0.142	0.115	0.113	0.132	0.127
Leduc(2,13,1)	0.150	**0.044**	0.084	0.065	0.453	0.127
Leduc(2,13,2)	0.150	**0.077**	0.127	0.089	1.091	0.128
Leduc(2,13,3)	0.150	**0.089**	0.113	0.090	1.108	0.117
FHP	10.640	**5.077**	6.153	6.142	97.313	8.334

Takeaways from these experiments:

- In practice, unsafe subgame solving often does pretty well!
- Safe KLSS is…
  - better than unsafe KLSS (unsurprisingly)
  - almost as good as common-knowledge subgame solving!

Liu, Fu, Fu, Yang, "Opponent-Limited Online Search for Imperfect Information Games", *ICML* 2023