

Stopping Criteria for Value Iteration for Concurrent Stochastic Reachability and Safety Games

Marta Grobelna¹, Jan Křetínský^{2,1}, Maximilian Weininger^{3,4,1}

¹Technical University of Munich

²Masaryk University Brno

³Ruhr University Bochum

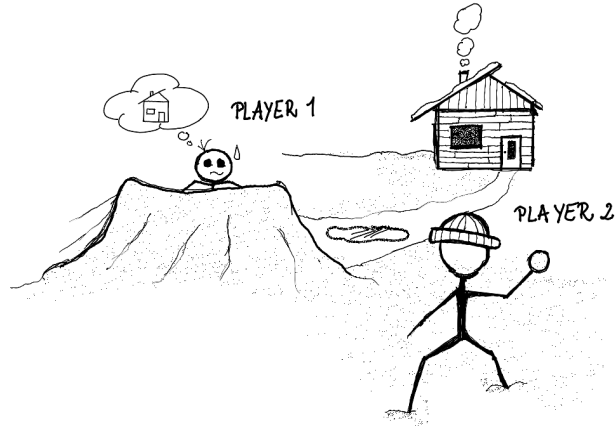
⁴Institute of Science and Technology Austria

Singapore, 25.06.2025

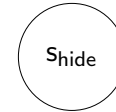
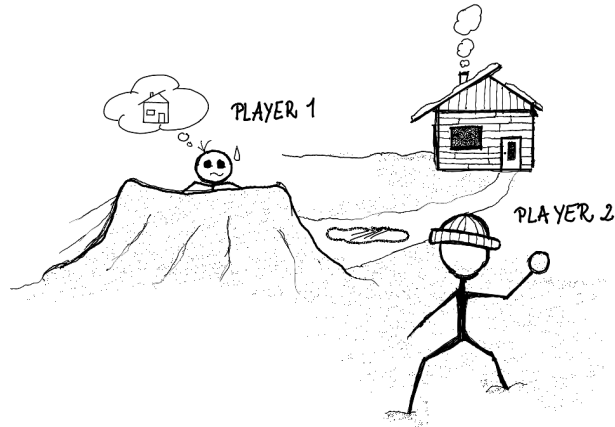


TUM Uhrenturm

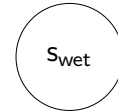
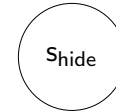
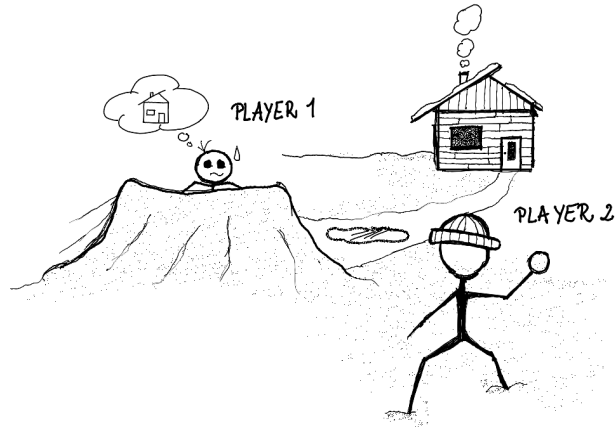
Concurrent Stochastic Reachability Game — Snow_Ball



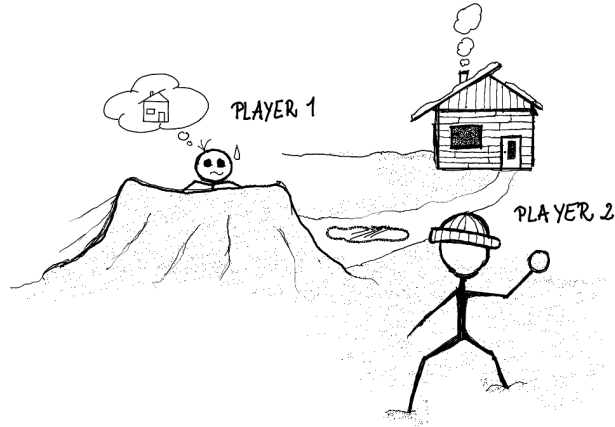
Concurrent Stochastic Reachability Game — Snow_Ball



Concurrent Stochastic Reachability Game — Snow_Ball



Concurrent Stochastic Reachability Game — Snow_Ball

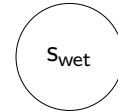
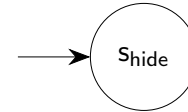
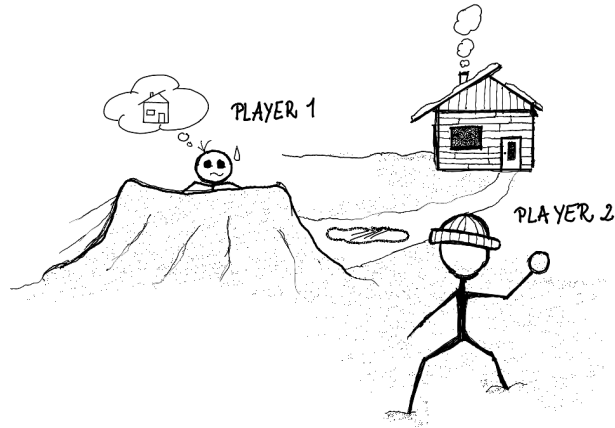


Shide

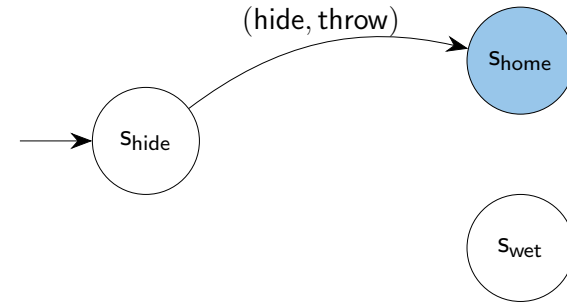
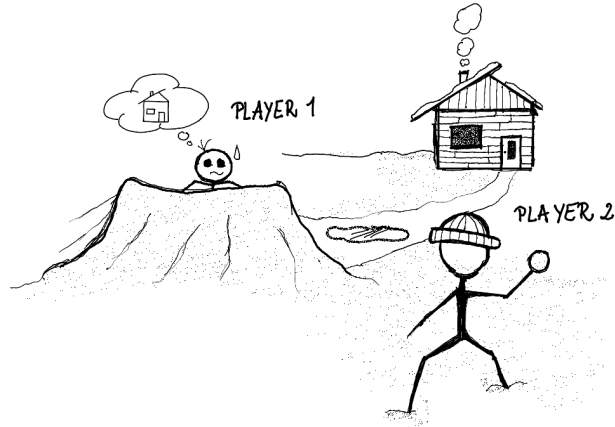
Shome

Swet

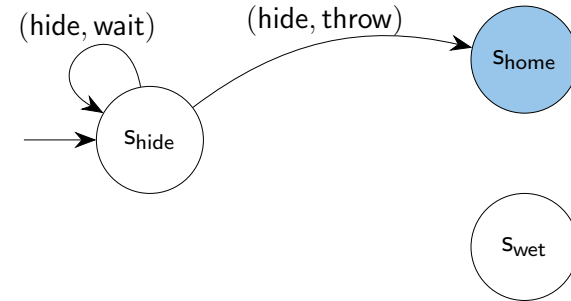
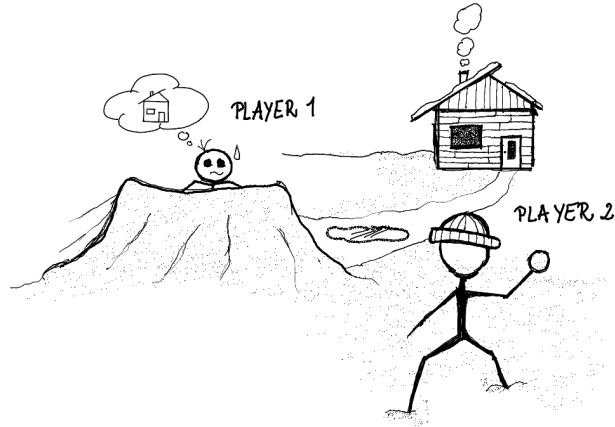
Concurrent Stochastic Reachability Game — Snow_Ball



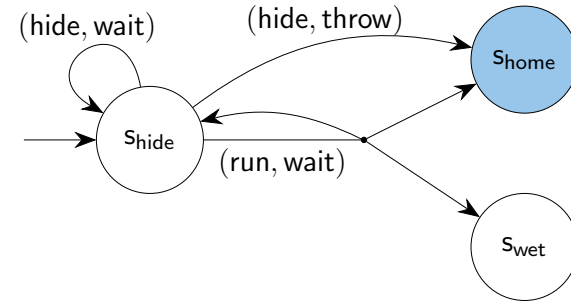
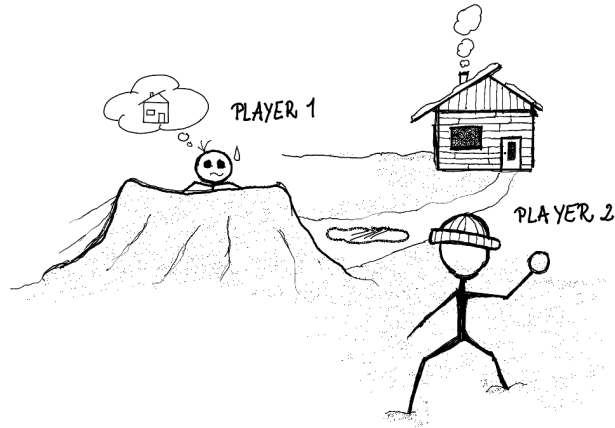
Concurrent Stochastic Reachability Game — Snow_Ball



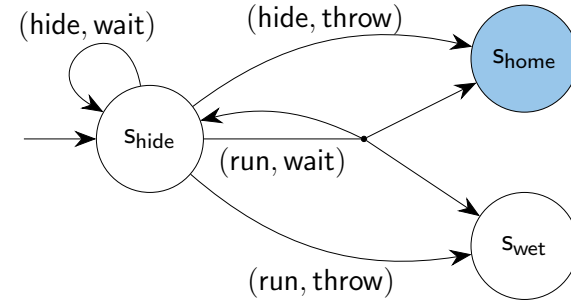
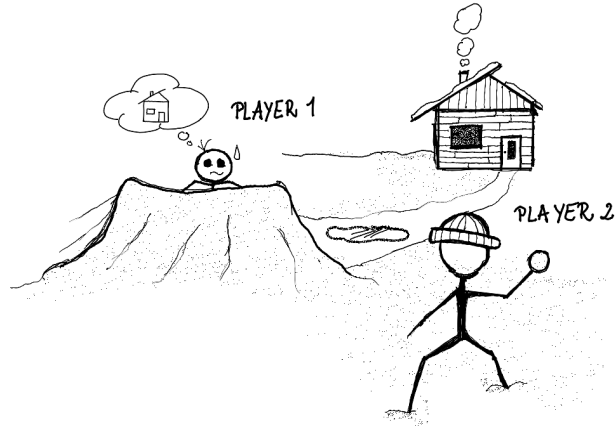
Concurrent Stochastic Reachability Game — Snow_Ball



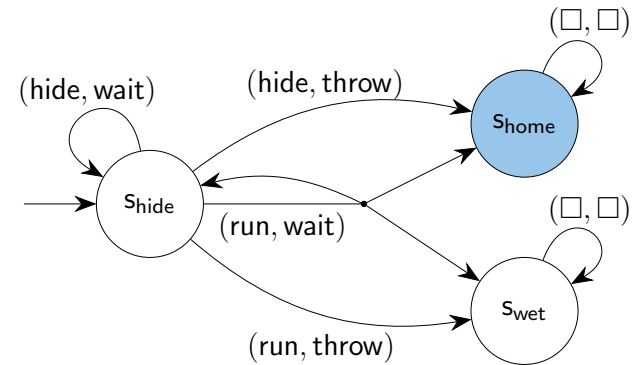
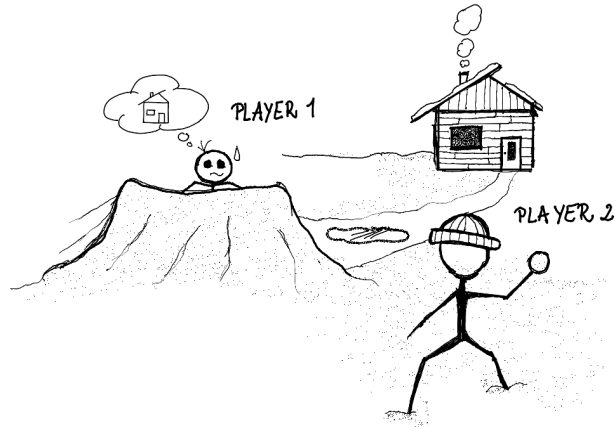
Concurrent Stochastic Reachability Game — Snow_Ball



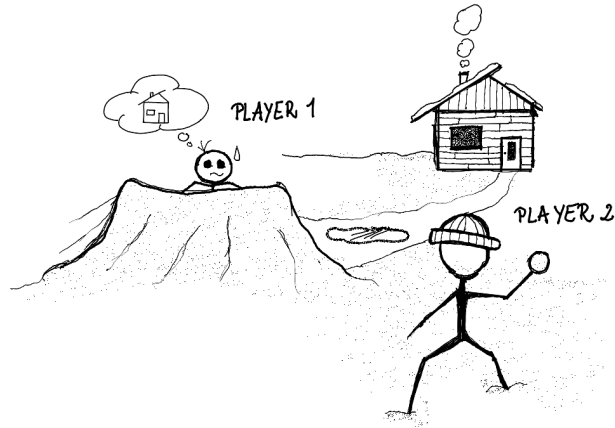
Concurrent Stochastic Reachability Game — Snow_Ball



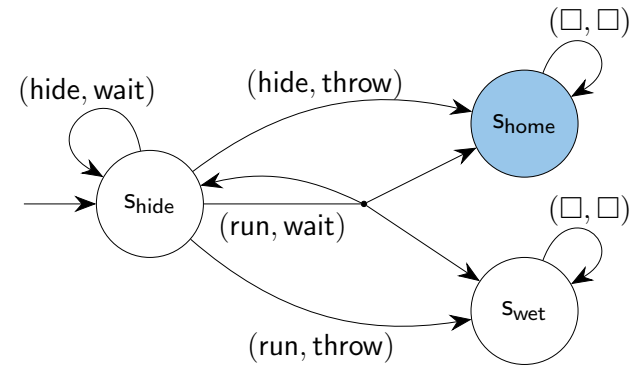
Concurrent Stochastic Reachability Game — Snow_Ball



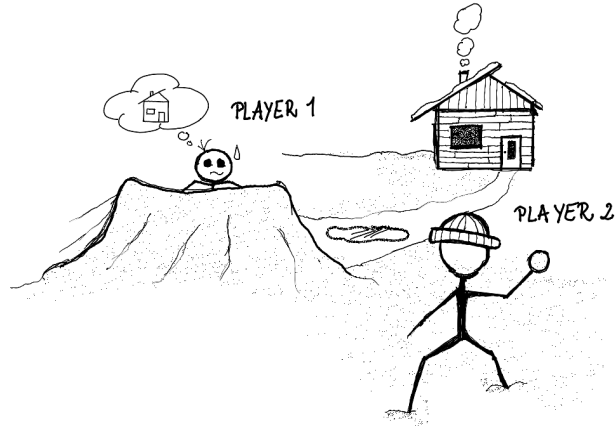
Concurrent Stochastic Reachability Game — Snow_Ball



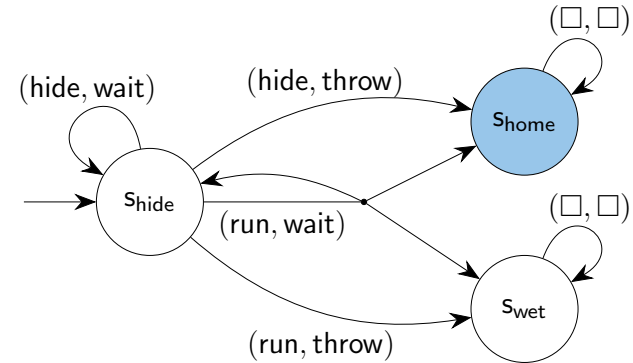
- Player 1: Maximizer (Reachability player)



Concurrent Stochastic Reachability Game — Snow_Ball



- Player 1: Maximizer (Reachability player)
- Player 2: Minimizer (Safety player)



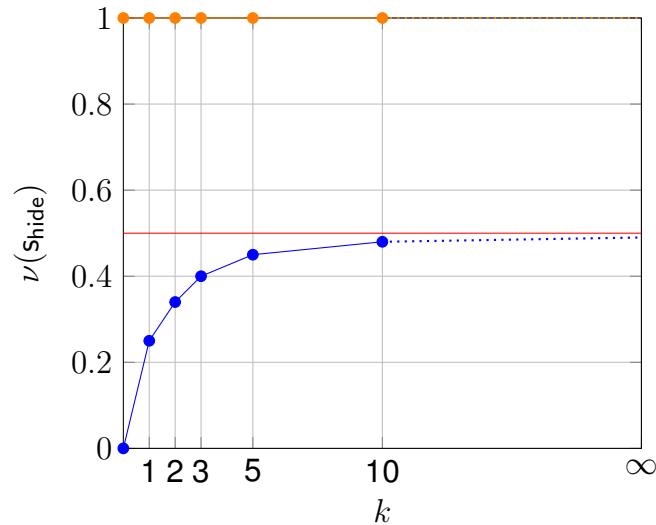
What is This Talk About?

What is This Talk About?

- Bounded Value Iteration for CSGs

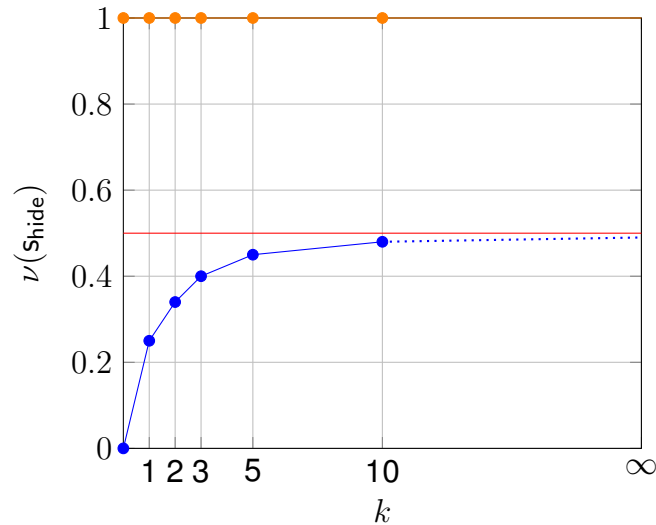
What is This Talk About?

- Bounded Value Iteration for CSGs
- Make non-convergent over-approximation converge



What is This Talk About?

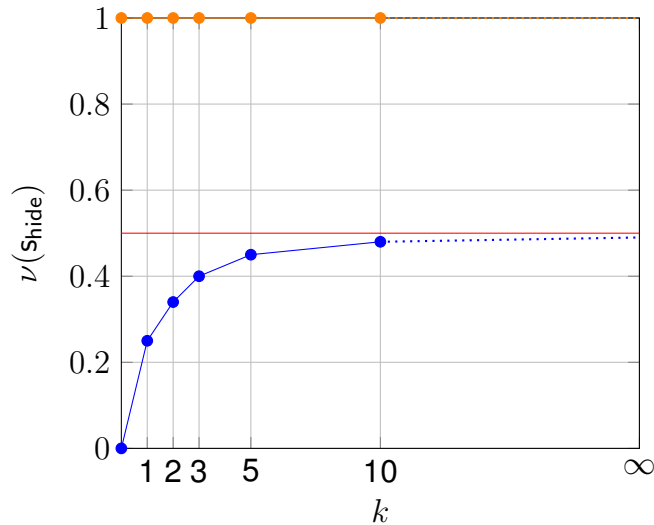
- Bounded Value Iteration for CSGs
- Make non-convergent over-approximation converge



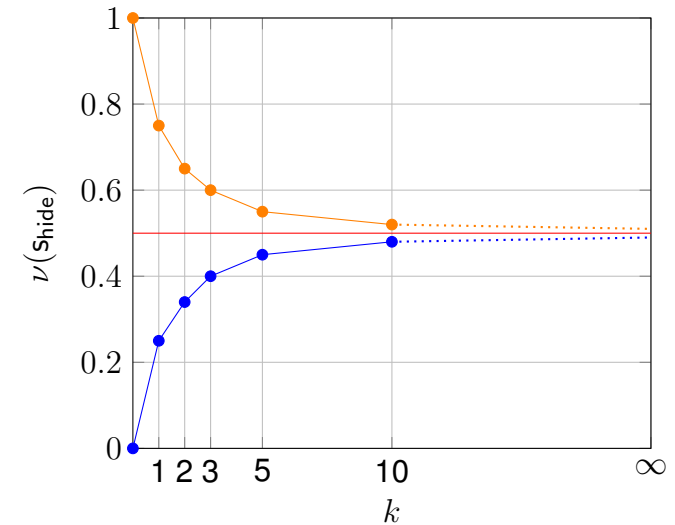
DEFLATING 

What is This Talk About?

- Bounded Value Iteration for CSGs
- Make non-convergent over-approximation converge

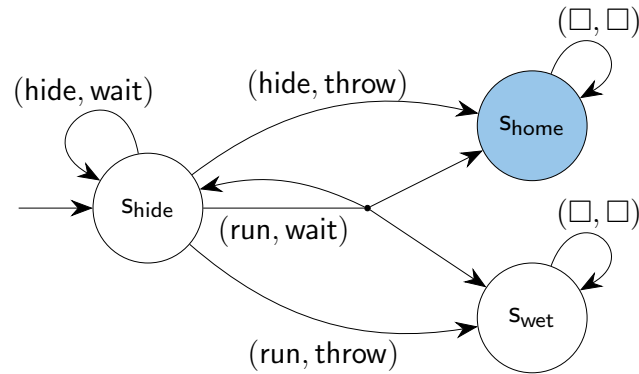


DEFLATING

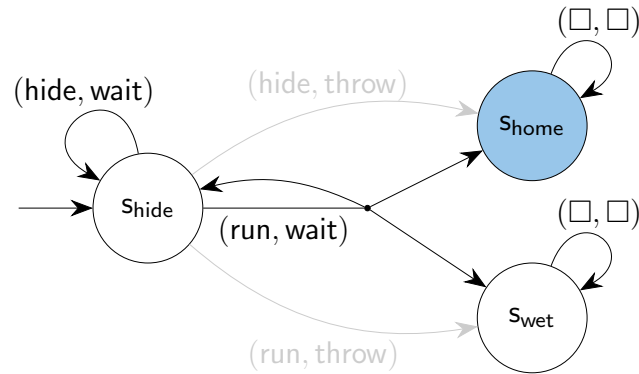


Bounded Value Iteration for Guarantees on the Approximation

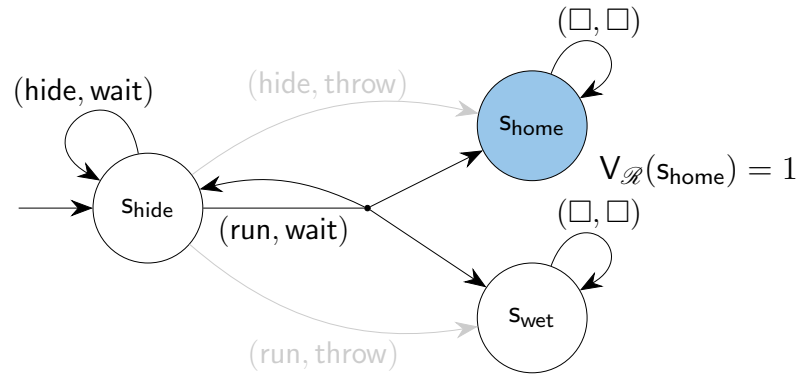
Bounded Value Iteration for Guarantees on the Approximation



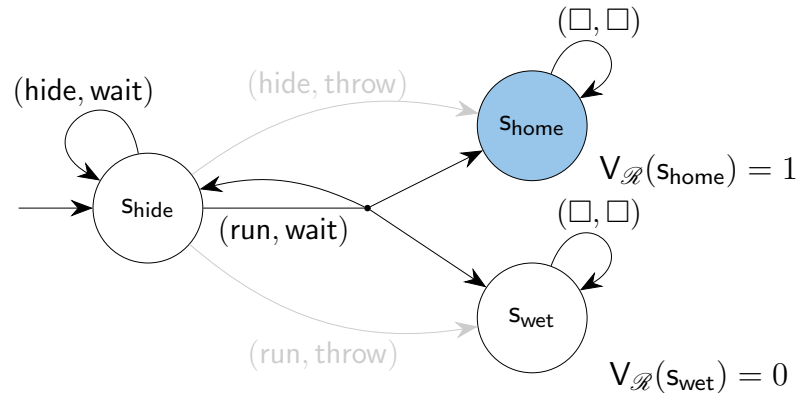
Bounded Value Iteration for Guarantees on the Approximation



Bounded Value Iteration for Guarantees on the Approximation

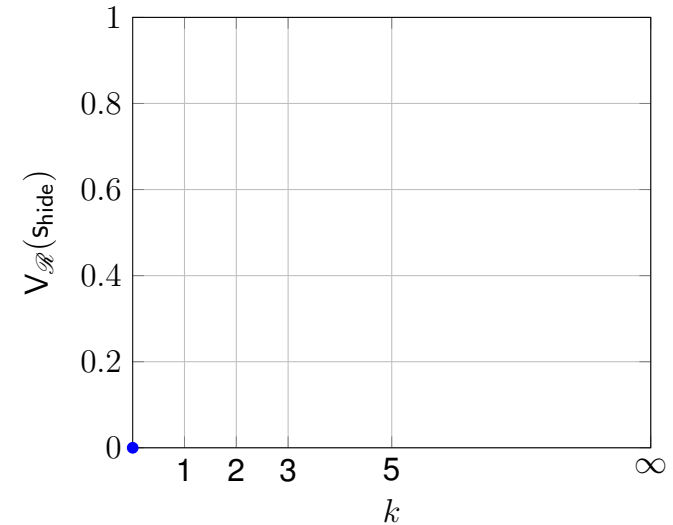
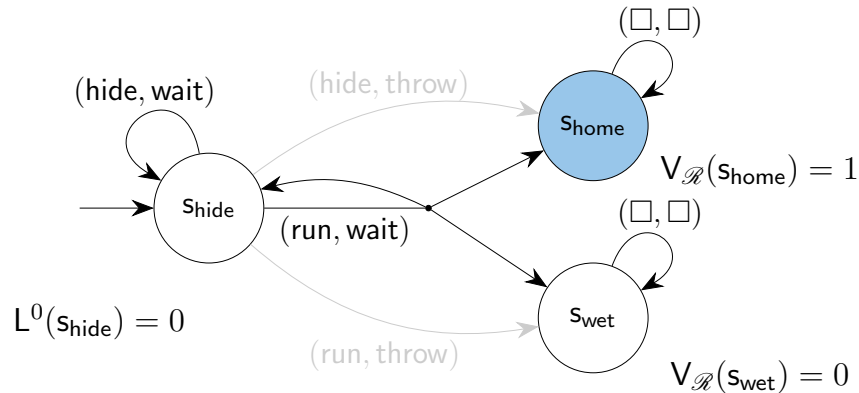


Bounded Value Iteration for Guarantees on the Approximation



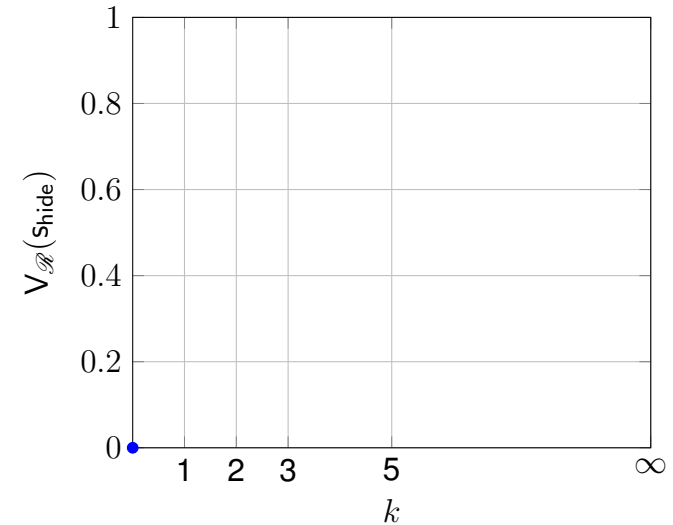
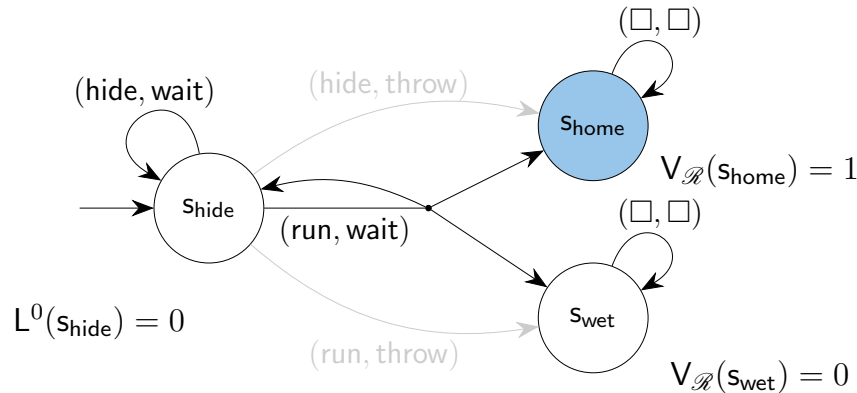
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with lowest possible value



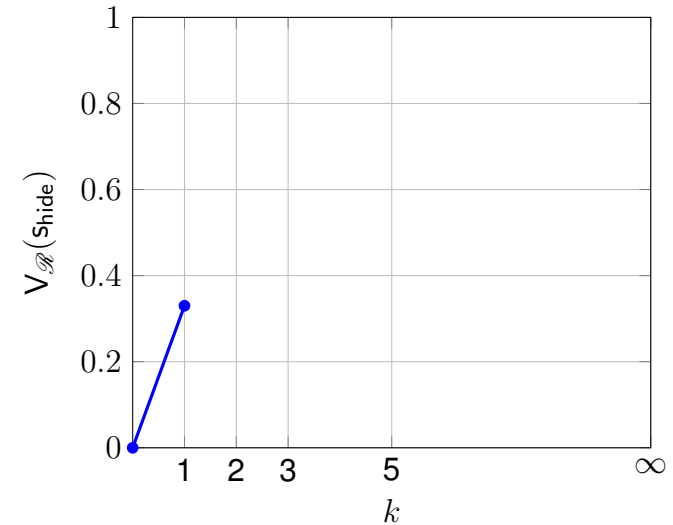
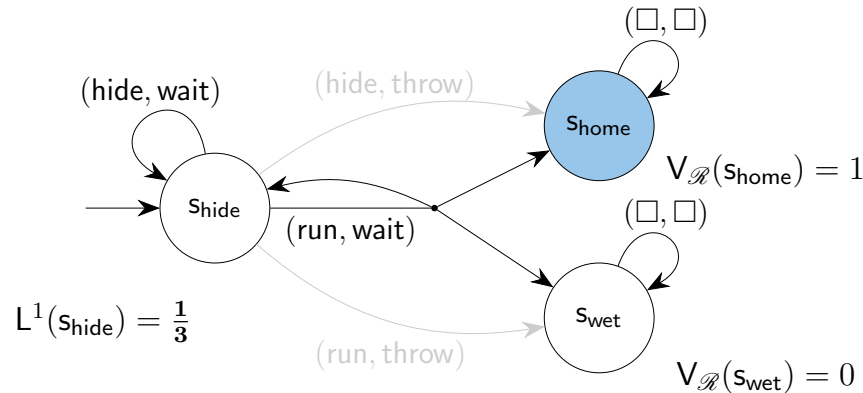
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with lowest possible value
- Iteratively update by back-propagating from target states



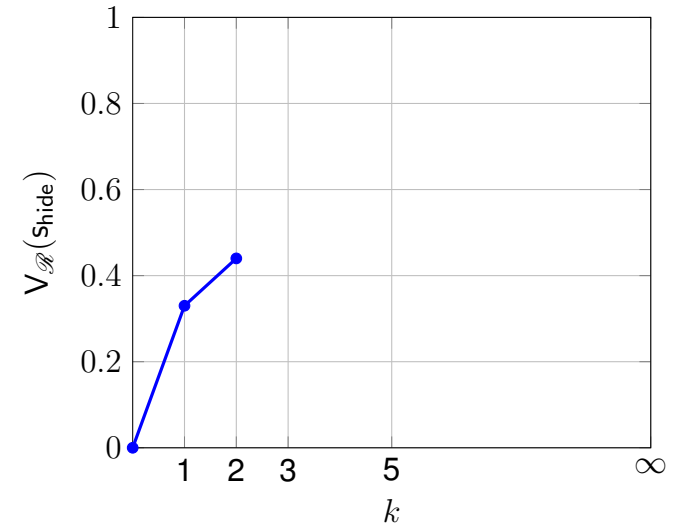
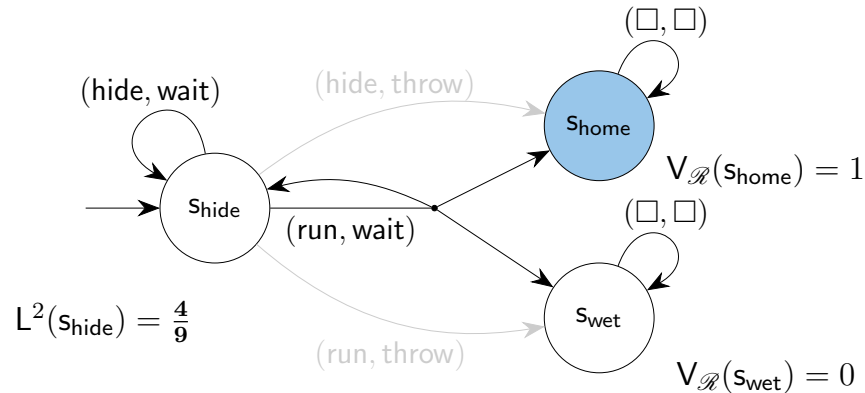
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with lowest possible value
- Iteratively update by back-propagating from target states



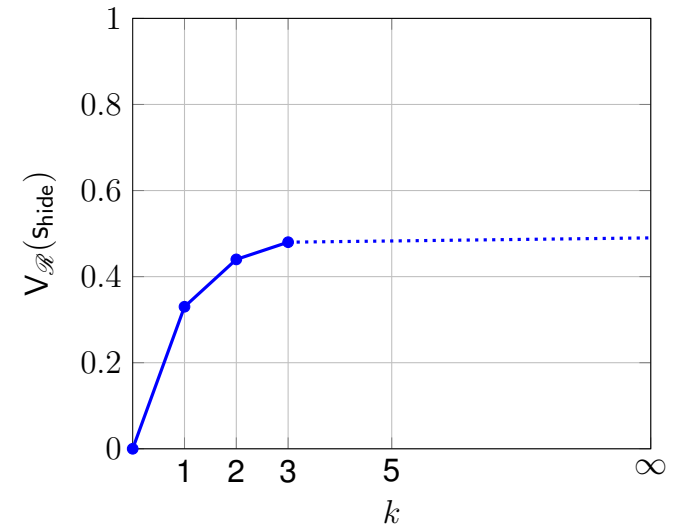
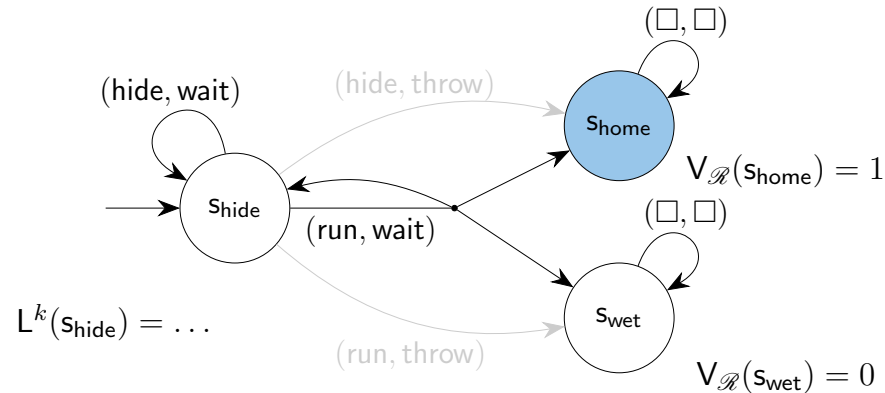
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with lowest possible value
- Iteratively update by back-propagating from target states



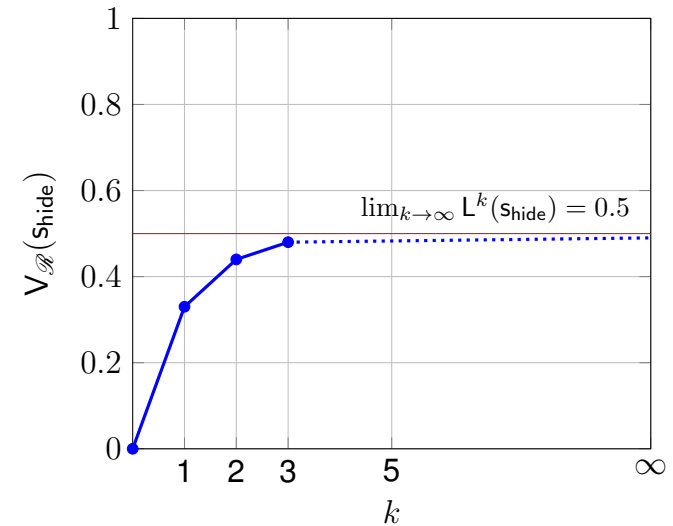
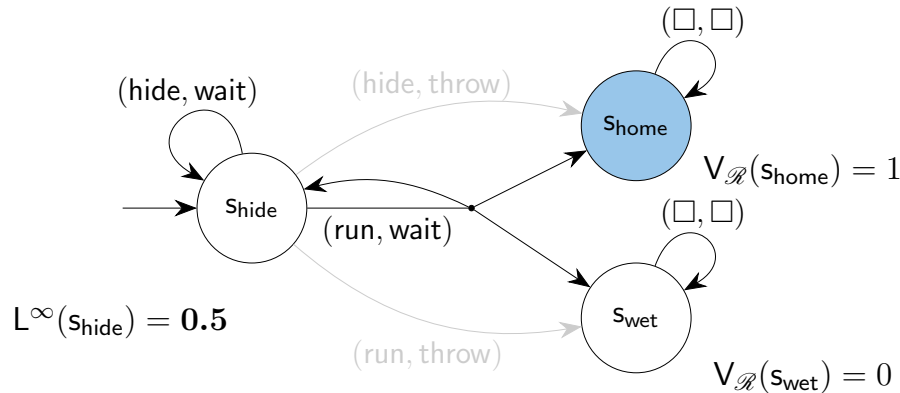
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with lowest possible value
- Iteratively update by back-propagating from target states



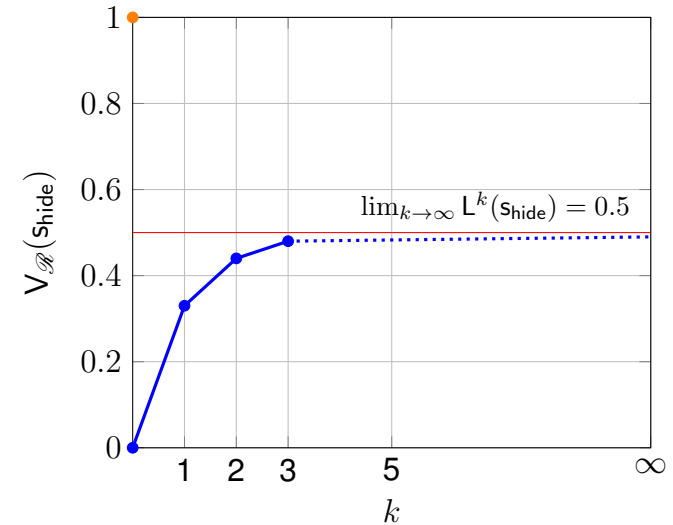
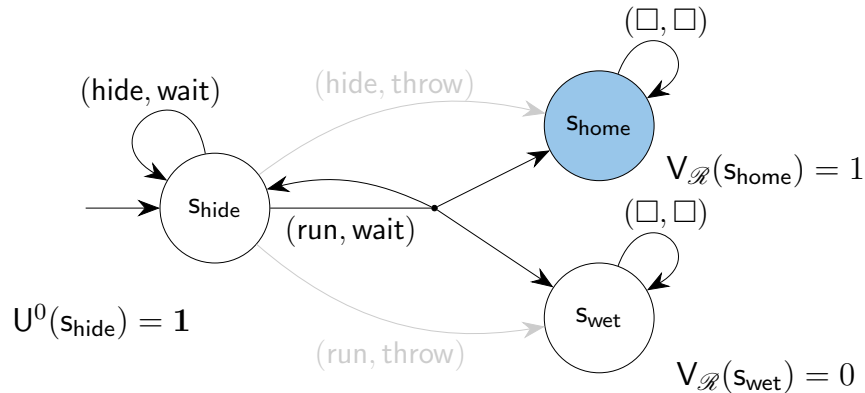
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with lowest possible value
- Iteratively update by back-propagating from target states



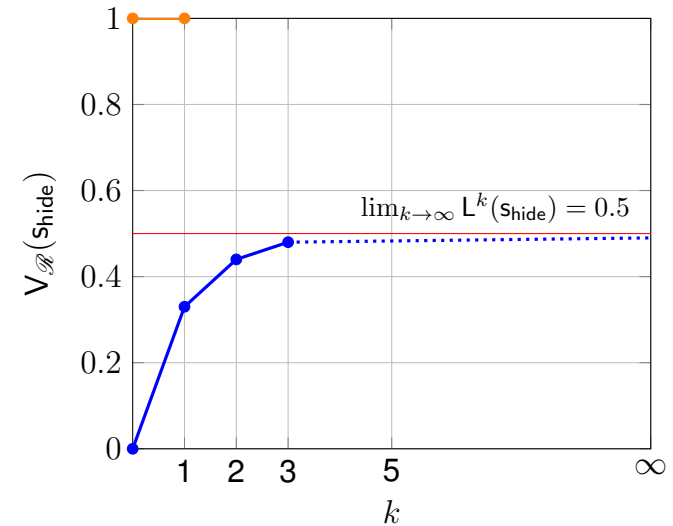
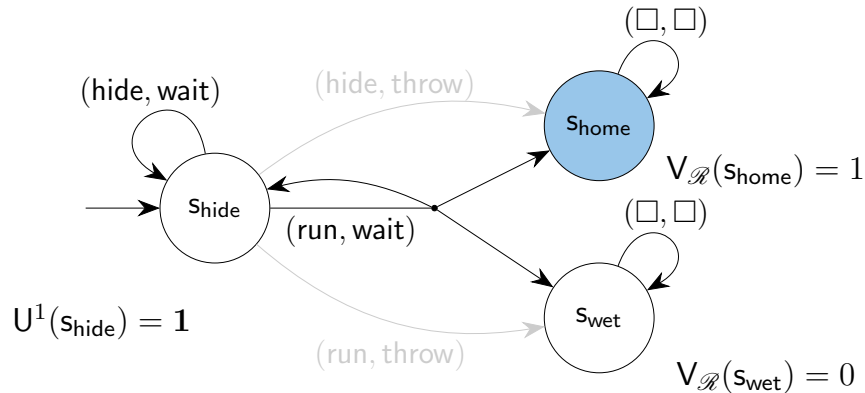
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with *highest* possible value
- Iteratively update by back-propagating from sink states



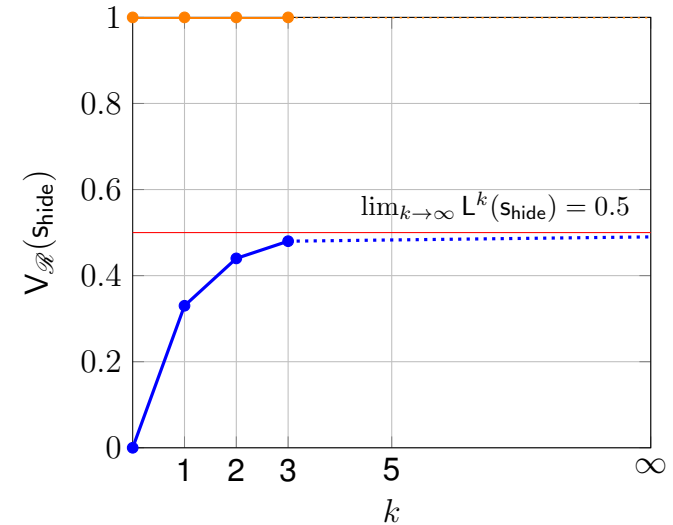
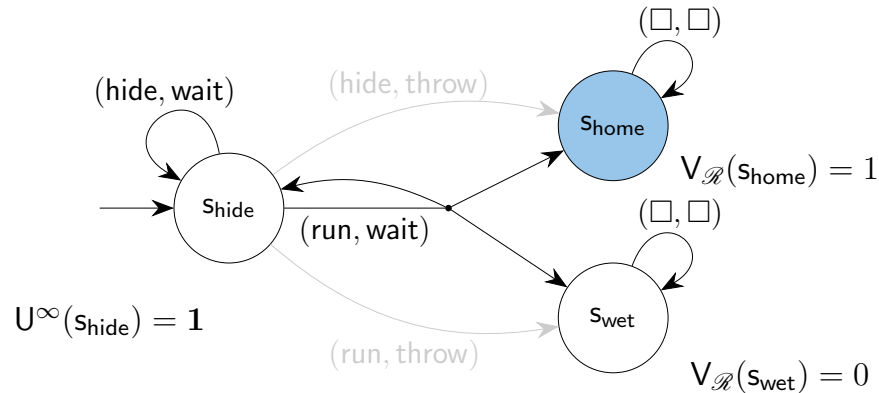
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with *highest* possible value
- Iteratively update by back-propagating from sink states



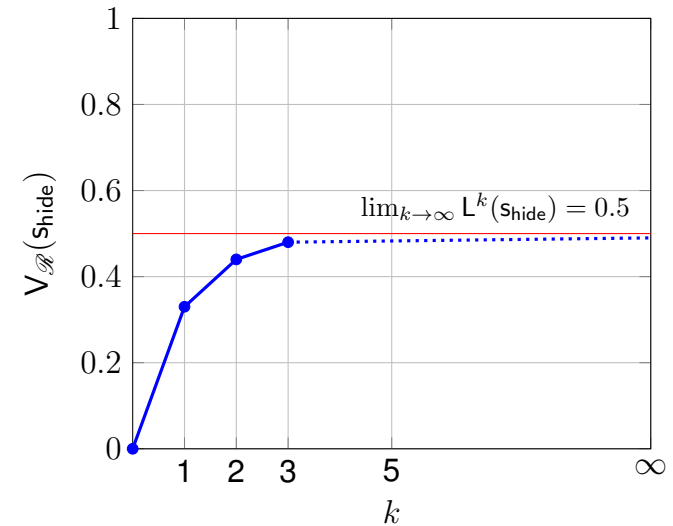
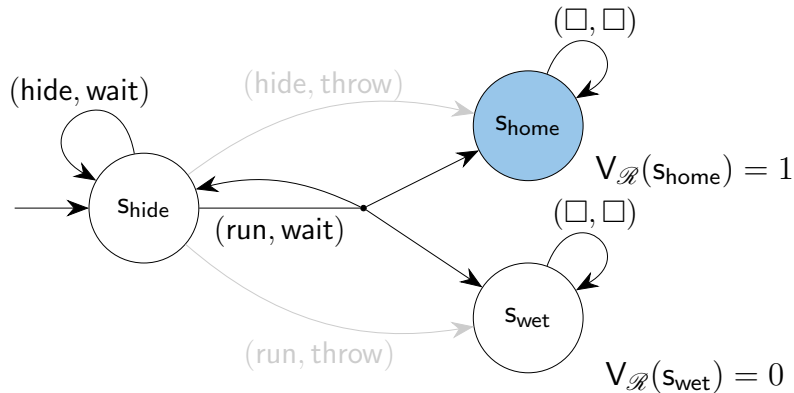
Bounded Value Iteration for Guarantees on the Approximation

- Initialize with *highest* possible value
- Iteratively update by back-propagating from sink states



Bounded Value Iteration for Guarantees on the Approximation

- Initialize with *highest* possible value
- Iteratively update by back-propagating from sink states
- **How to ensure convergence of the upper bound?**



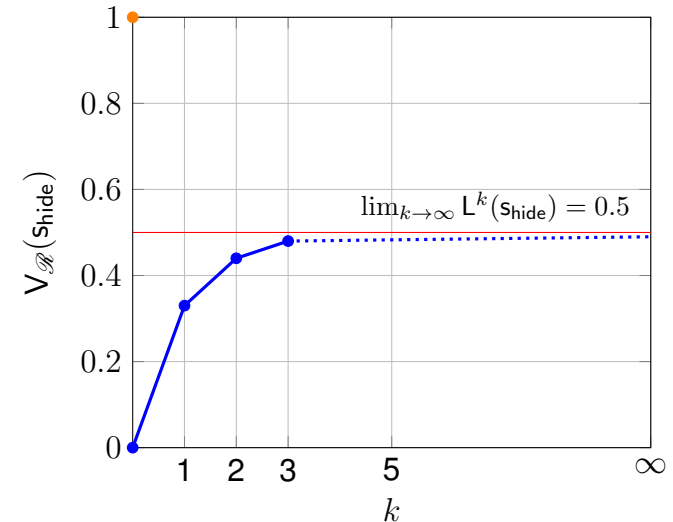
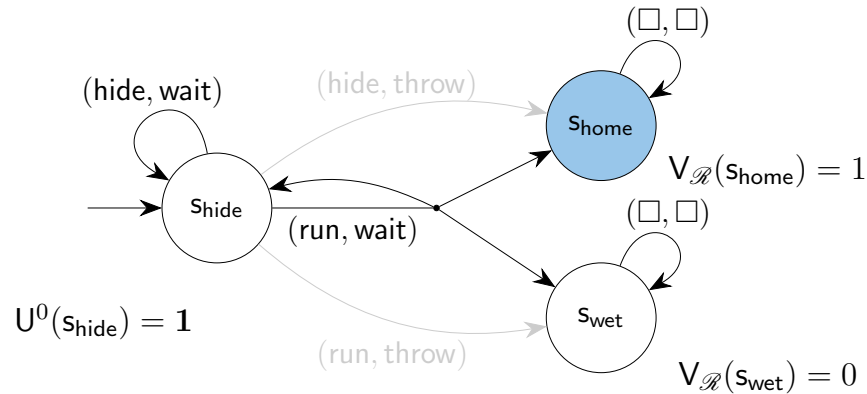
Deflating Ensures Convergence of the Upper Bound

Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value

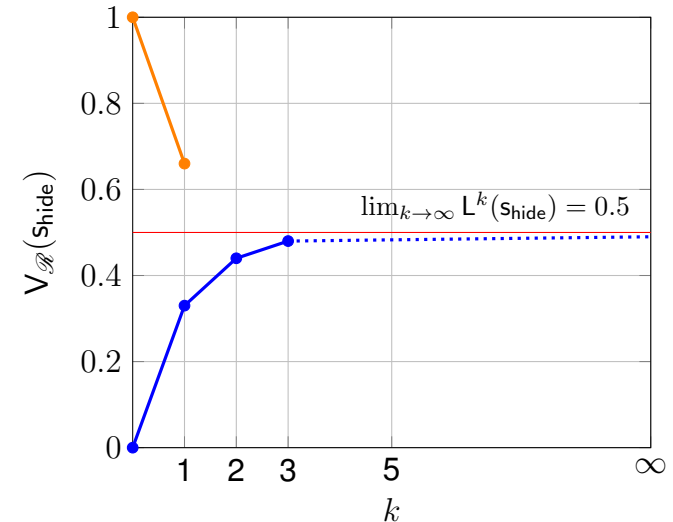
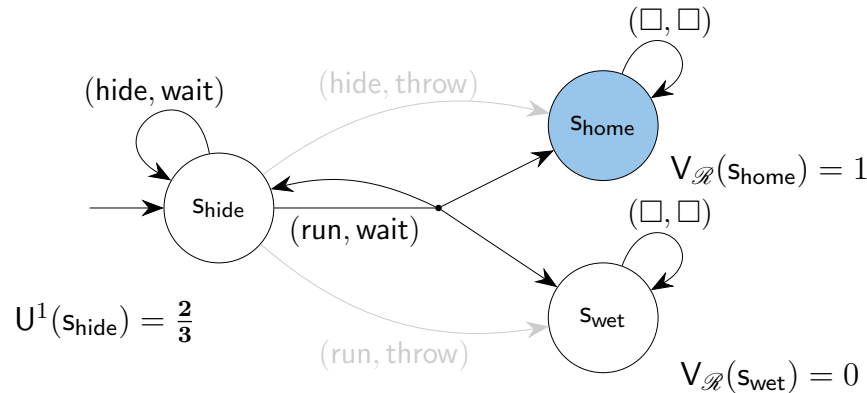
Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}



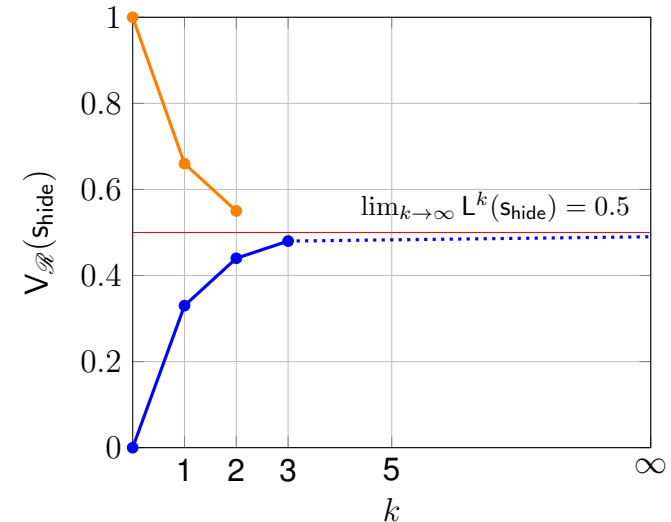
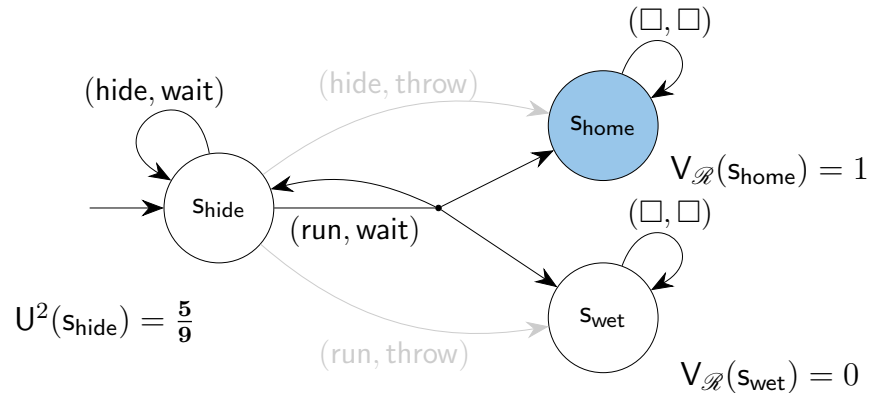
Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}



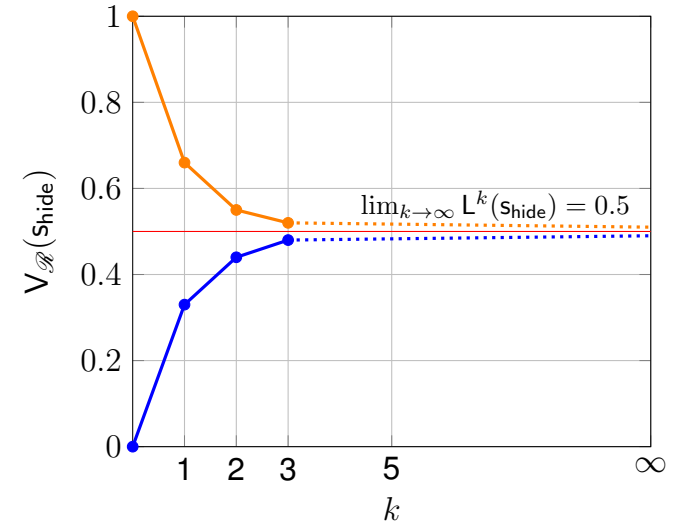
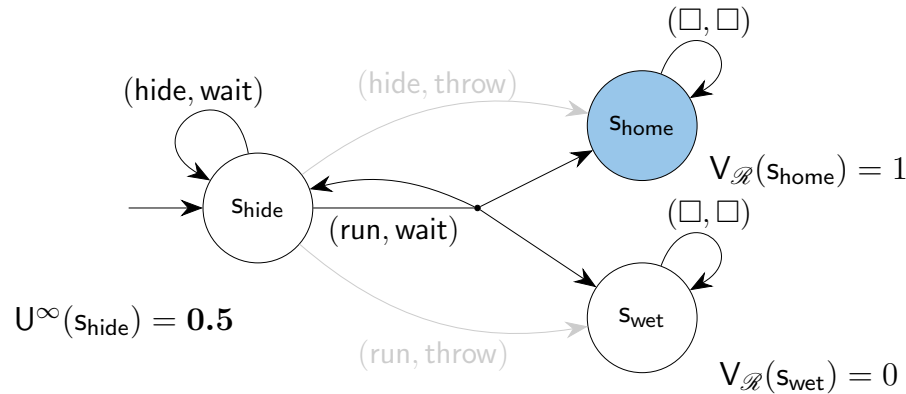
Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}



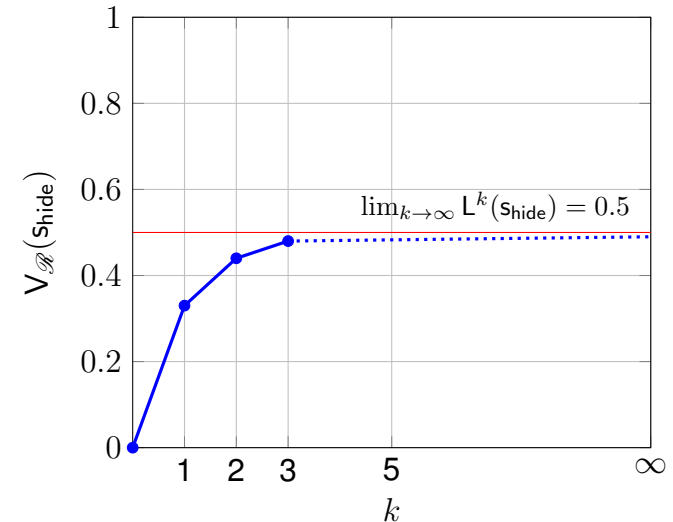
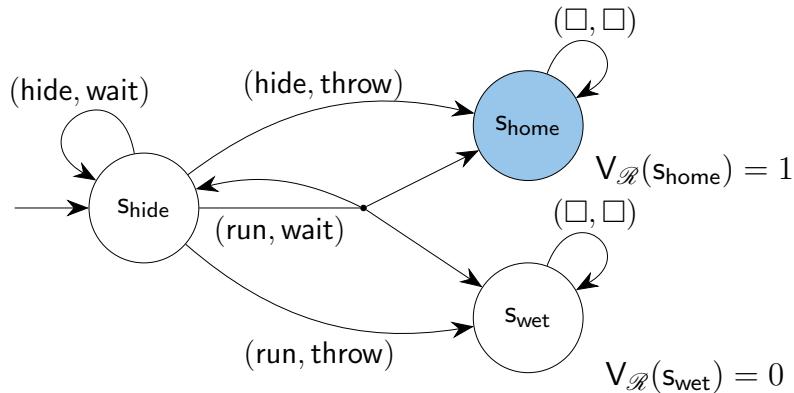
Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}



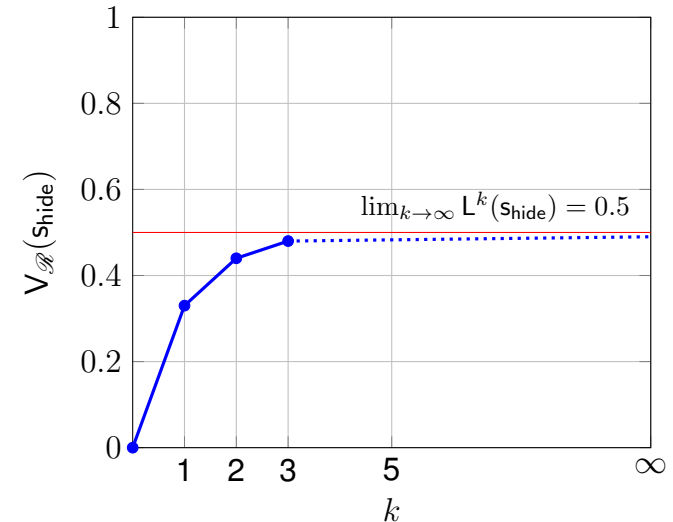
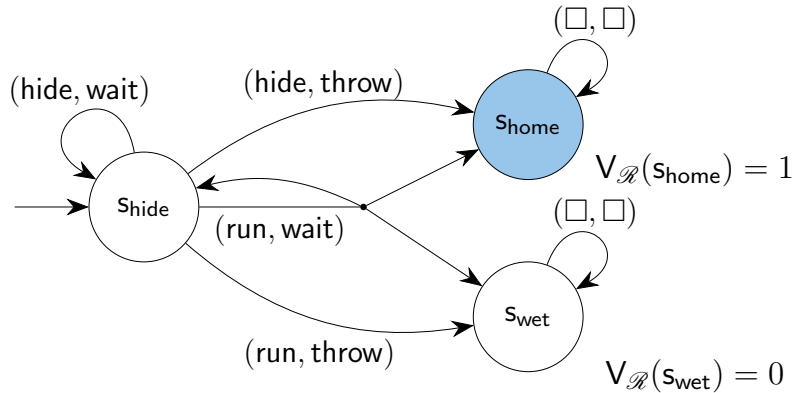
Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}



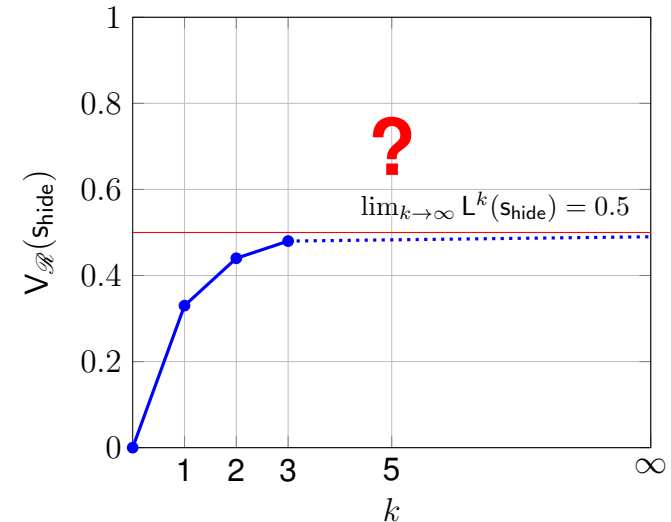
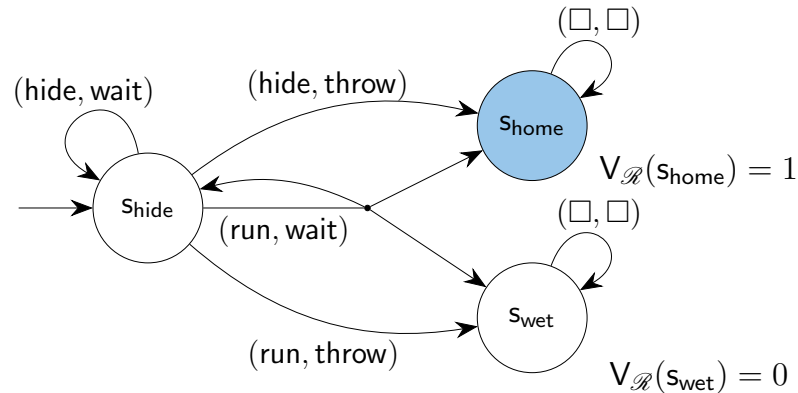
Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}
- What is the *leaving action* now?

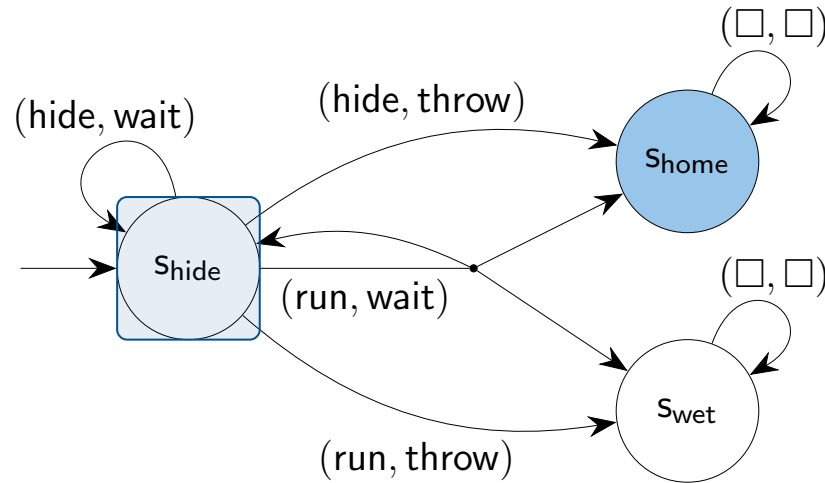


Deflating Ensures Convergence of the Upper Bound

- **Deflate:** Select *leaving* action attaining *highest* value
 - **Leaving:** For all opponent's actions leave s_{hide}
- What is the *leaving action* now?
- How to write this down?

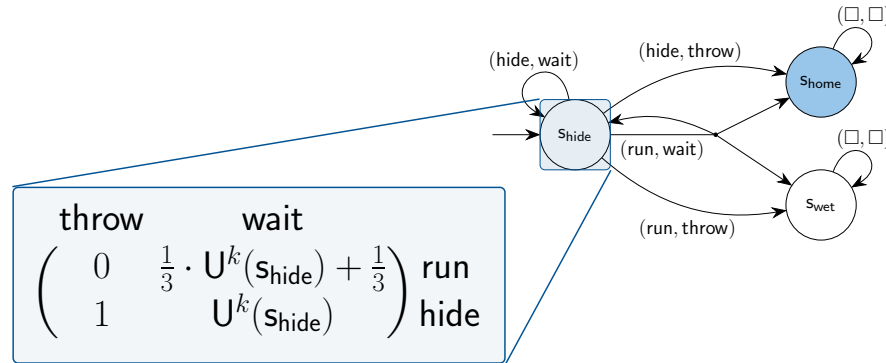


To Estimate Next Update Solve the Matrix Game



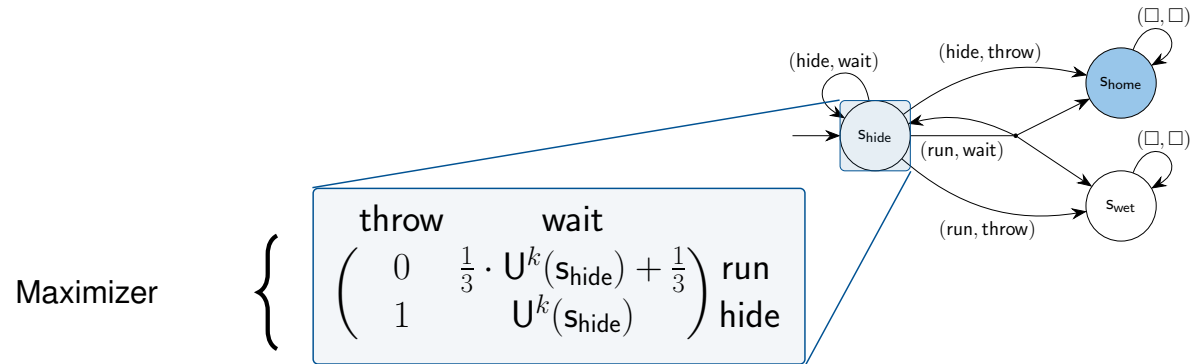
To Estimate Next Update Solve the Matrix Game

- At each state: *Solve* a two-player zero-sum **matrix game**



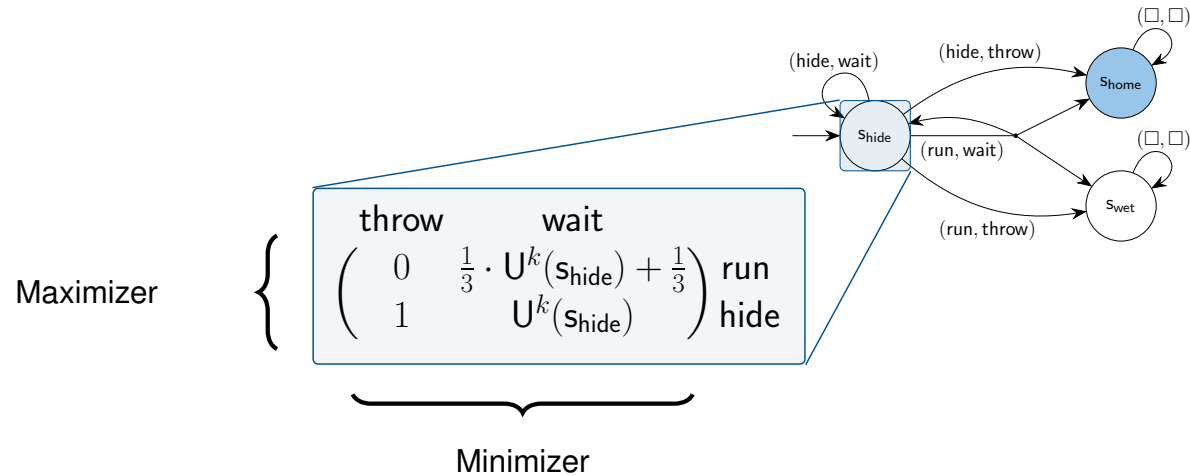
To Estimate Next Update Solve the Matrix Game

- At each state: *Solve* a two-player zero-sum **matrix game**



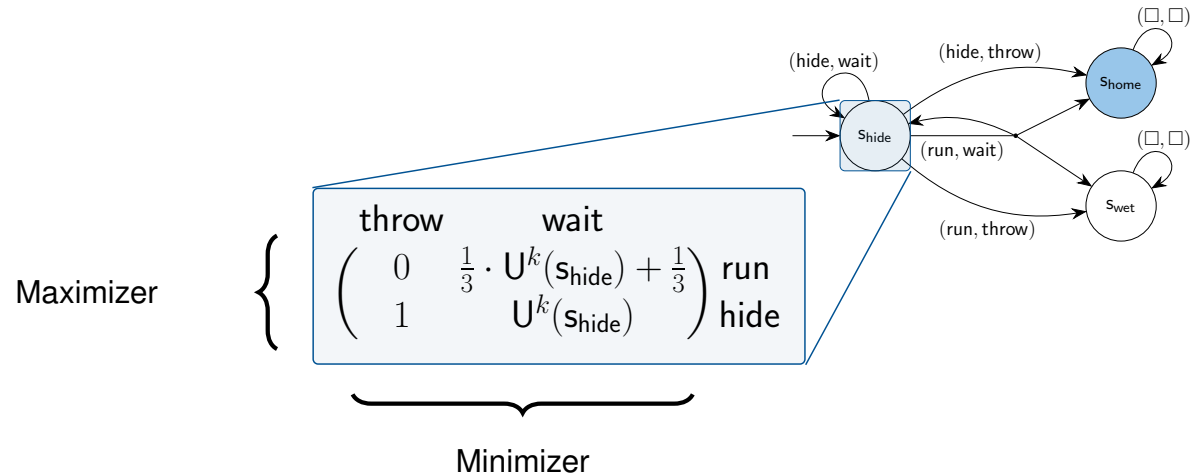
To Estimate Next Update Solve the Matrix Game

- At each state: *Solve* a two-player zero-sum **matrix game**



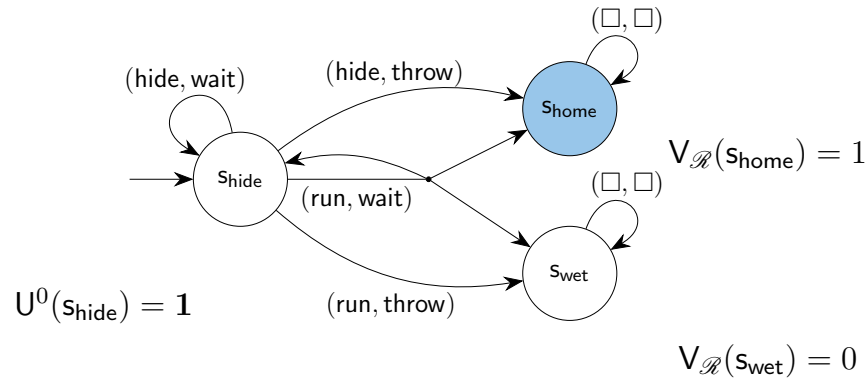
To Estimate Next Update Solve the Matrix Game

- At each state: *Solve* a two-player zero-sum **matrix game**
- Players choose **distributions over actions—strategies**

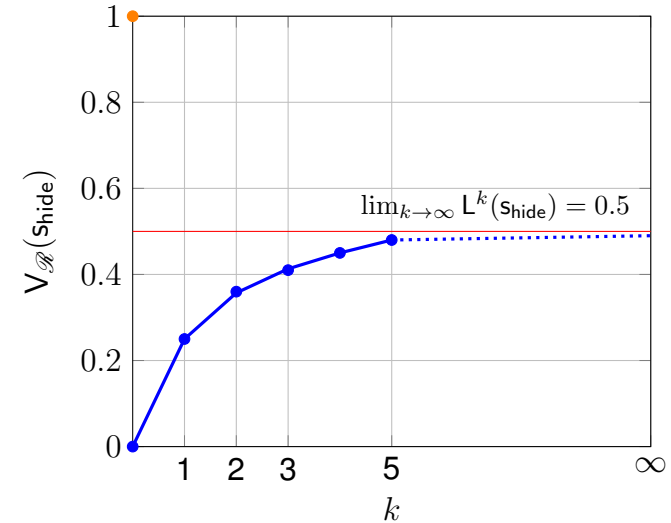


Deflating for CSGs is Hard

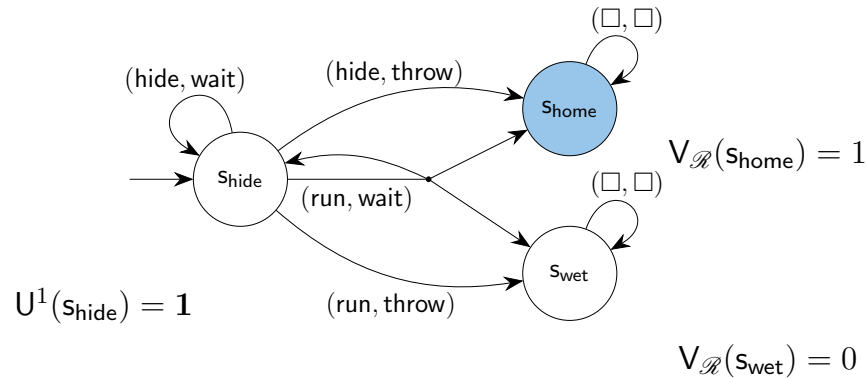
Deflating for CSGs is Hard



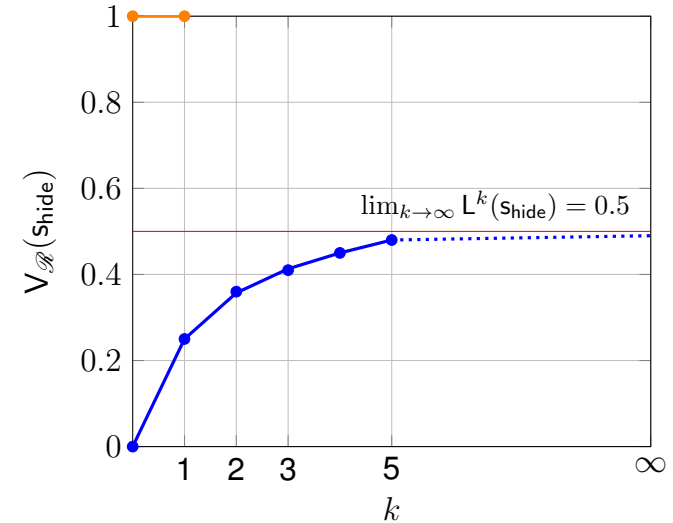
throw	wait	
$\begin{pmatrix} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & \frac{1}{3} \\ 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
run	hide	



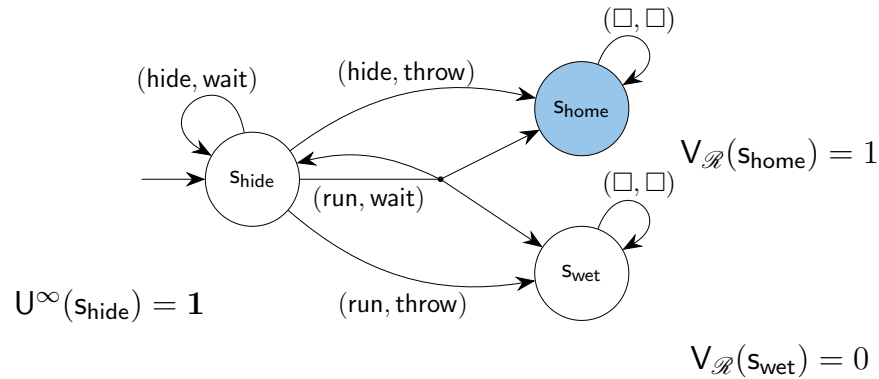
Deflating for CSGs is Hard



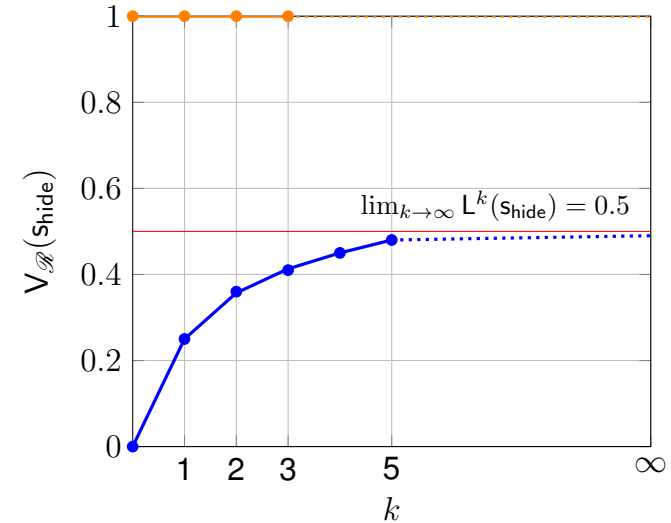
$$\begin{matrix} \text{throw} & \text{wait} \\ \begin{pmatrix} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix} & \begin{matrix} \text{run} \\ \text{hide} \end{matrix} \end{matrix}$$



Deflating for CSGs is Hard

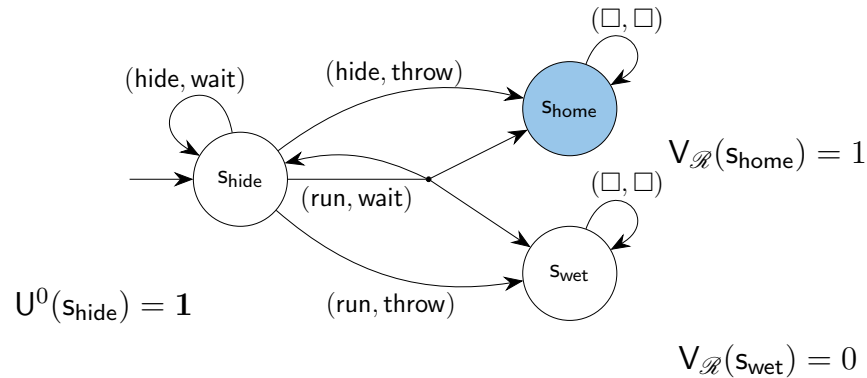


throw	wait	
$\begin{pmatrix} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix}$	run	hide

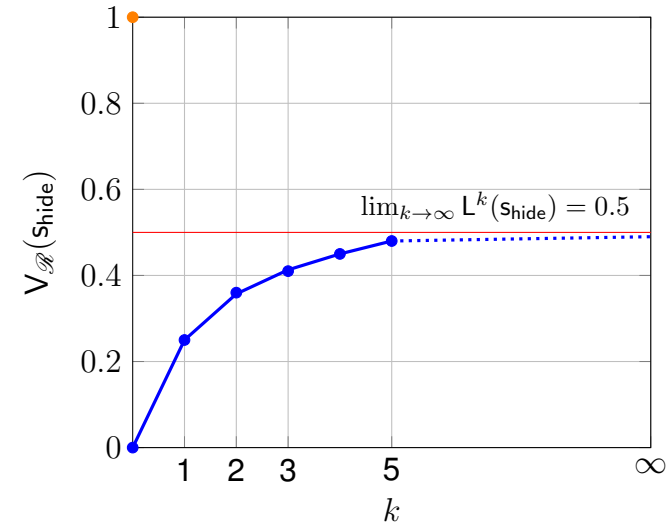


Deflating for CSGs is Hard

- **Deflate:** Select *leaving strategy*...

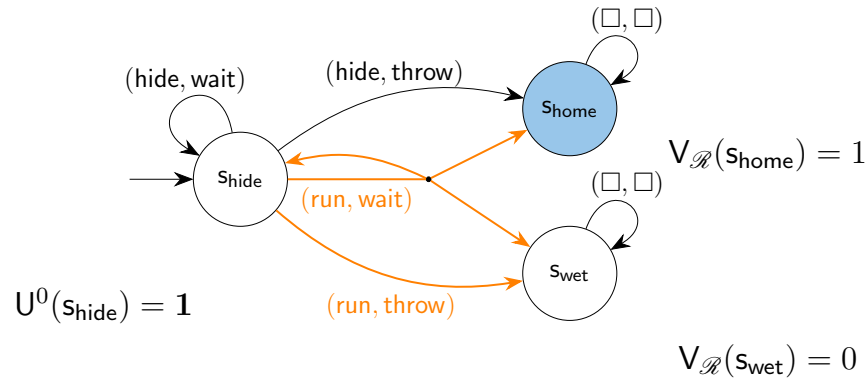


$$\begin{pmatrix} \text{throw} & \text{wait} \\ 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix} \begin{matrix} \text{run} \\ \text{hide} \end{matrix}$$

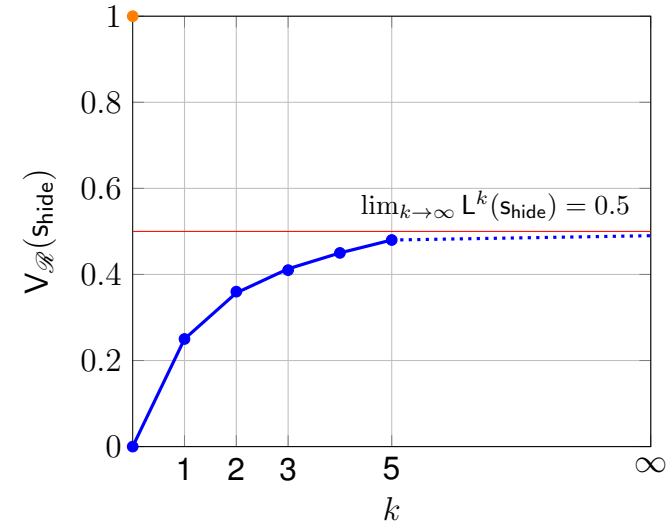


Deflating for CSGs is Hard

- **Deflate:** Select *leaving strategy*...
- Use $\{\text{run} \mapsto 1\}$ to deflate?

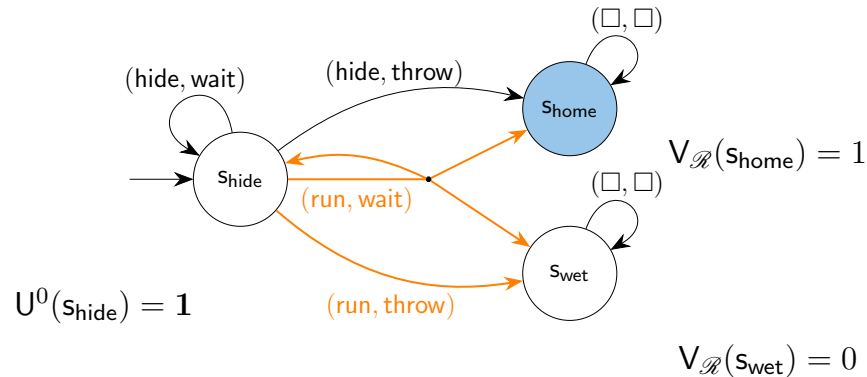


$$\begin{pmatrix} \text{throw} & \text{wait} \\ 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix} \begin{matrix} \text{run} \\ \text{hide} \end{matrix}$$

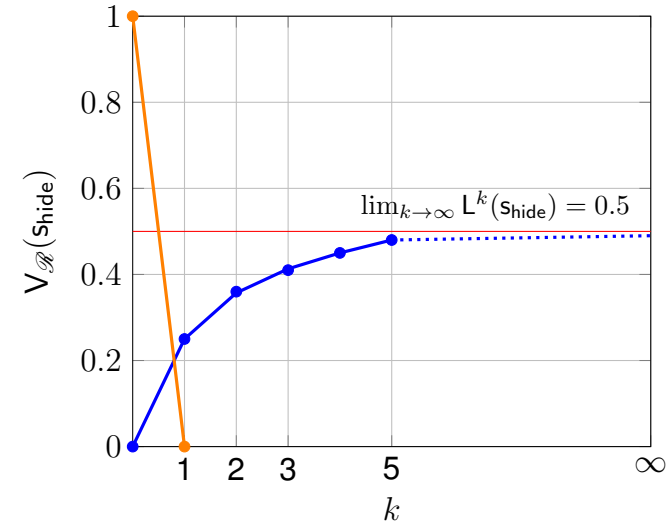


Deflating for CSGs is Hard

- **Deflate:** Select *leaving strategy*...
- Use $\{\text{run} \mapsto 1\}$ to deflate? $\Rightarrow$ Attains value $0 < 0.5$

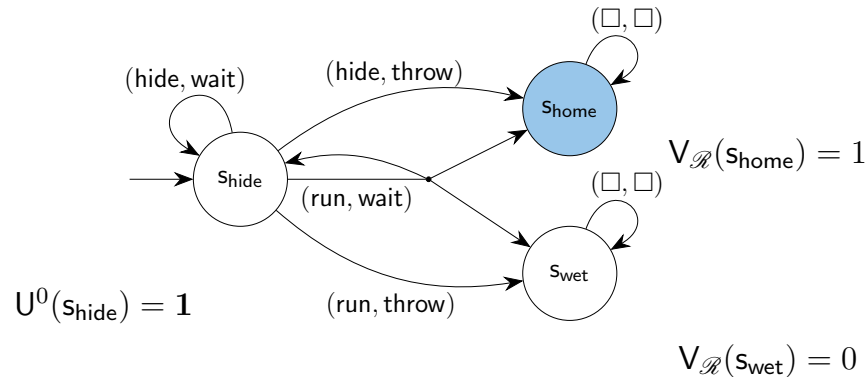


$$\begin{pmatrix} \text{throw} & \text{wait} \\ 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix} \begin{matrix} \text{run} \\ \text{hide} \end{matrix}$$

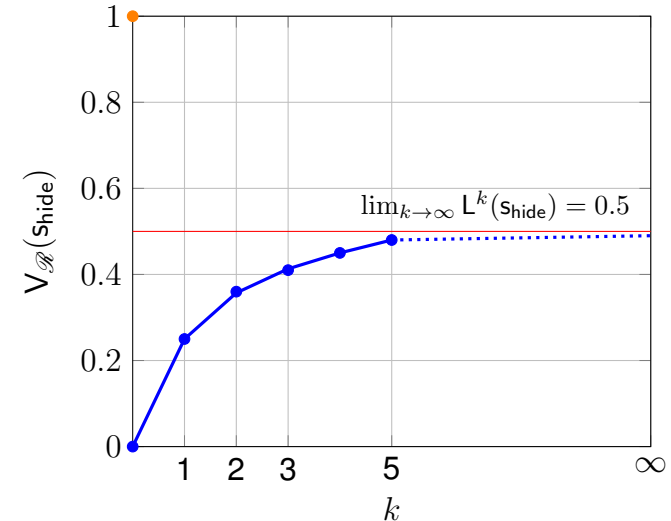


Deflating for CSGs is Hard

- **Deflate:** Select *leaving strategy*...
- Use $\{\text{run} \mapsto 1\}$ to deflate? $\Rightarrow$ Attains value $0 < 0.5$
- Consider $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ to deflate?

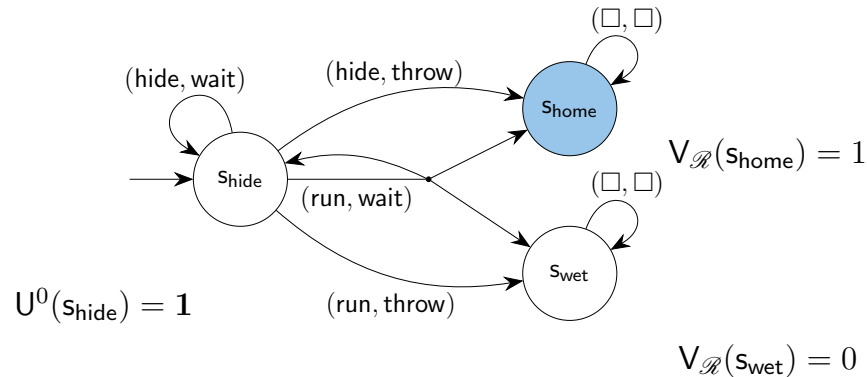


$$\begin{pmatrix} \text{throw} & \text{wait} \\ 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix} \begin{matrix} \text{run} \\ \text{hide} \end{matrix}$$

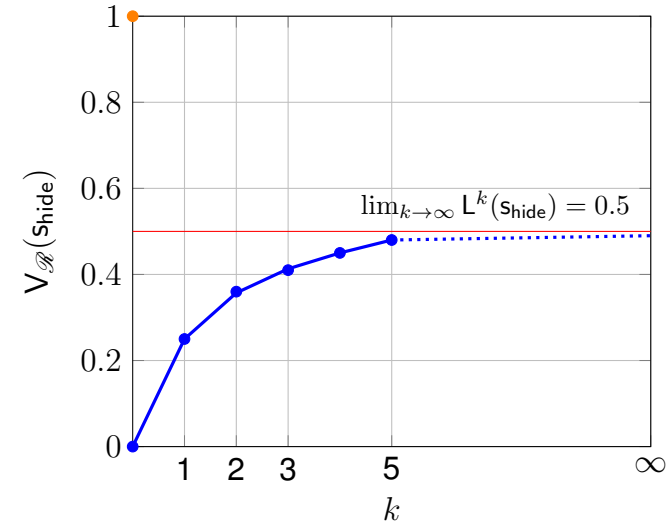


Deflating for CSGs is Hard

- **Deflate:** Select *leaving strategy*... **attaining highest value**
- Use $\{\text{run} \mapsto 1\}$ to deflate? $\Rightarrow$ Attains value $0 < 0.5$
- Consider $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ to deflate?



$$\begin{array}{cc} \text{throw} & \text{wait} \\ \begin{pmatrix} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix} & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Domination is Needed to Identify Optimal Strategies

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy?

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy? **NO!**
 - **Supremum is misleading here!**

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy? **NO!**
 - **Supremum is misleading here!**
- **Dominant strategy:** *For all* opponent's strategies better or equal and at least for one *strictly* better

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy? **NO!**
 - **Supremum is misleading here!**
- **Dominant strategy:** *For all* opponent's strategies better or equal and at least for one *strictly* better
 - $\{\text{run} \mapsto 1\} \prec \{\text{hide} \mapsto 1\}$

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy? **NO!**
 - **Supremum is misleading here!**
- **Dominant strategy:** *For all* opponent's strategies better or equal and at least for one *strictly* better
 - $\{\text{run} \mapsto 1\} \prec \{\text{hide} \mapsto 1\}$
- **Extend to sets of strategies**

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Domination is Needed to Identify Optimal Strategies

- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy? **NO!**
 - **Supremum is misleading here!**
- **Dominant strategy:** *For all* opponent's strategies better or equal and at least for one *strictly* better
 - $\{\text{run} \mapsto 1\} \prec \{\text{hide} \mapsto 1\}$
- **Extend to sets of strategies**
 - $\{\text{hide} \mapsto 1\} \preceq \text{RunPositive}?$

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$

Domination is Needed to Identify Optimal Strategies

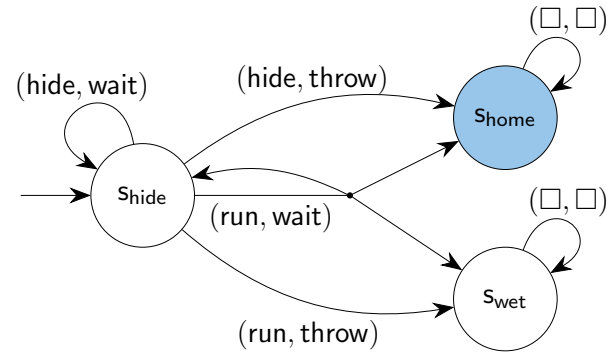
- **Optimal strategy:** Supremum over all strategies
 - $\{\text{hide} \mapsto 1\}$ attains 1
 - Supremum in $\text{RunPositive} := \{\{\text{run} \mapsto \epsilon, \text{hide} \mapsto 1 - \epsilon\} \mid \epsilon > 0\}$ also attains 1
 - So we have optimal leaving strategy? **NO!**
 - **Supremum is misleading here!**
- **Dominant strategy:** *For all* opponent's strategies better or equal and at least for one *strictly* better
 - $\{\text{run} \mapsto 1\} \prec \{\text{hide} \mapsto 1\}$
- **Extend to sets of strategies**
 - $\{\text{hide} \mapsto 1\} \preceq \text{RunPositive}$? **NO!**

$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$

Hazardous Strategies Hinder Convergence of Upper Bound

Hazardous Strategies Hinder Convergence of Upper Bound

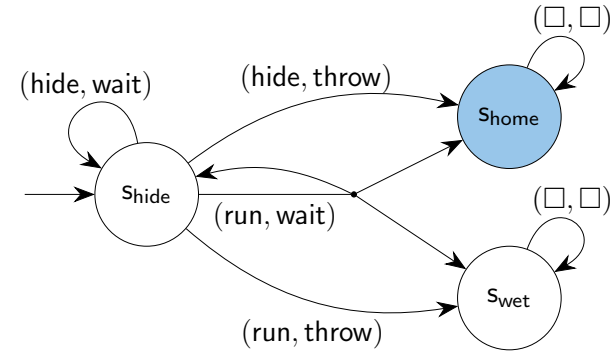
$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$



Hazardous Strategies Hinder Convergence of Upper Bound

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$

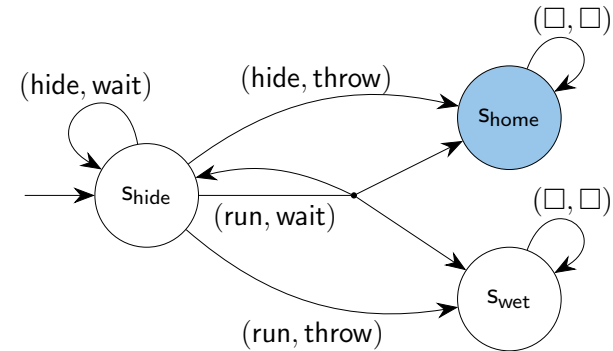
- $\{\text{hide} \mapsto \mathbf{1}\}$ is a *hazardous strategy*



Hazardous Strategies Hinder Convergence of Upper Bound

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$

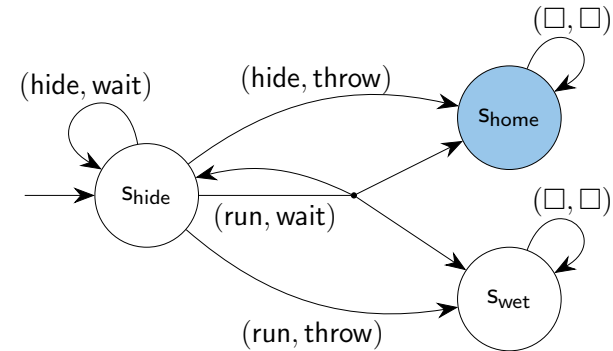
- $\{\text{hide} \mapsto 1\}$ is a *hazardous strategy*
 - Non-leaving: For $\{\text{wait} \mapsto 1\}$ we stay in s_{hide}



Hazardous Strategies Hinder Convergence of Upper Bound

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$

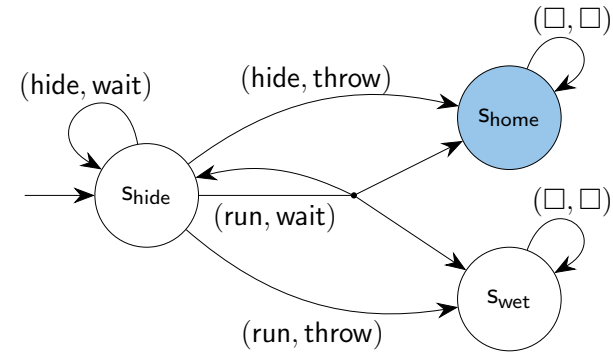
- $\{\text{hide} \mapsto 1\}$ is a *hazardous strategy*
 - Non-leaving: For $\{\text{wait} \mapsto 1\}$ we stay in s_{hide}
 - Optimal: $\mathcal{R} \setminus \{\text{hide} \mapsto 1\} \preceq \{\text{hide} \mapsto 1\}$



Hazardous Strategies Hinder Convergence of Upper Bound

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$

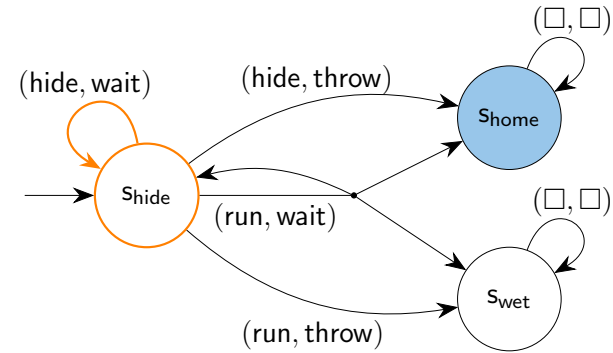
- $\{\text{hide} \mapsto 1\}$ is a *hazardous strategy*
 - Non-leaving: For $\{\text{wait} \mapsto 1\}$ we stay in s_{hide}
 - Optimal: $\mathcal{R} \setminus \{\text{hide} \mapsto 1\} \preceq \{\text{hide} \mapsto 1\}$
 - All leaving strategies are dominated:
RunPositive $\prec \{\text{hide} \mapsto 1\}$



Hazardous Strategies Hinder Convergence of Upper Bound

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$

- $\{\text{hide} \mapsto 1\}$ is a *hazardous strategy*
 - Non-leaving: For $\{\text{wait} \mapsto 1\}$ we stay in s_{hide}
 - Optimal: $\mathcal{R} \setminus \{\text{hide} \mapsto 1\} \preceq \{\text{hide} \mapsto 1\}$
 - All leaving strategies are dominated:
RunPositive $\prec \{\text{hide} \mapsto 1\}$

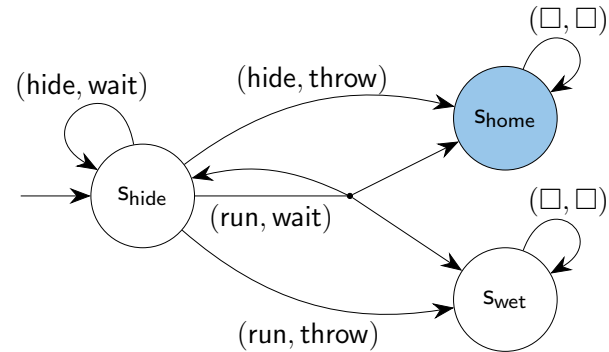


Value of Sub-matrix Game Yields Exit Value

Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies

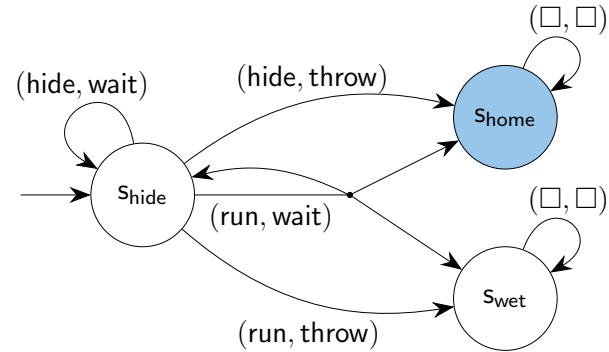
	throw	wait	
$\begin{pmatrix} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{pmatrix}$			run
			hide



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$

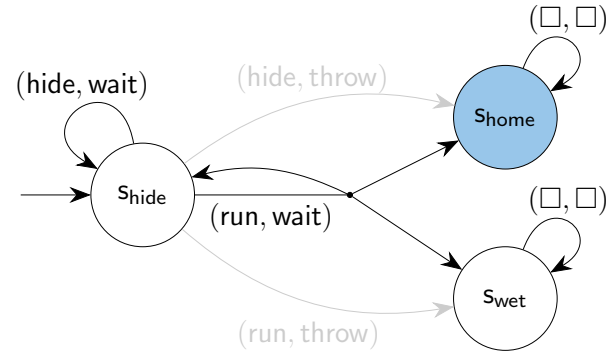
$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \mathbf{1} + \frac{1}{3} \\ 1 & \mathbf{1} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$

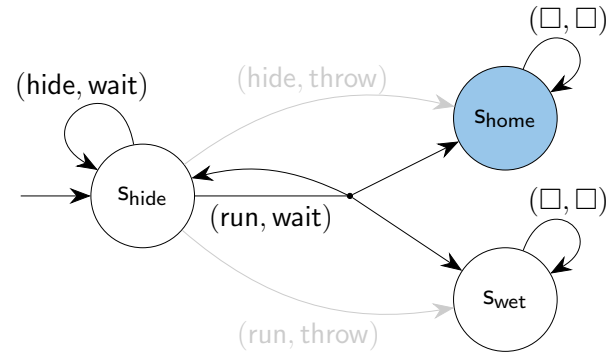
$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$

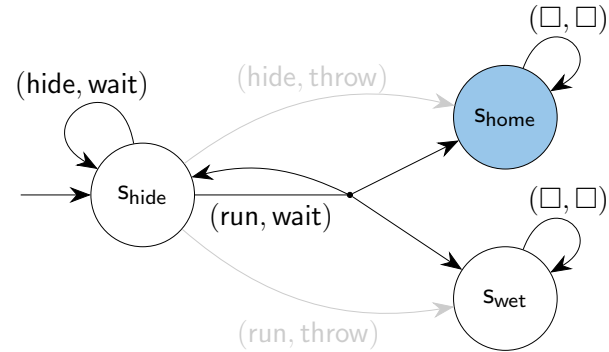
$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$

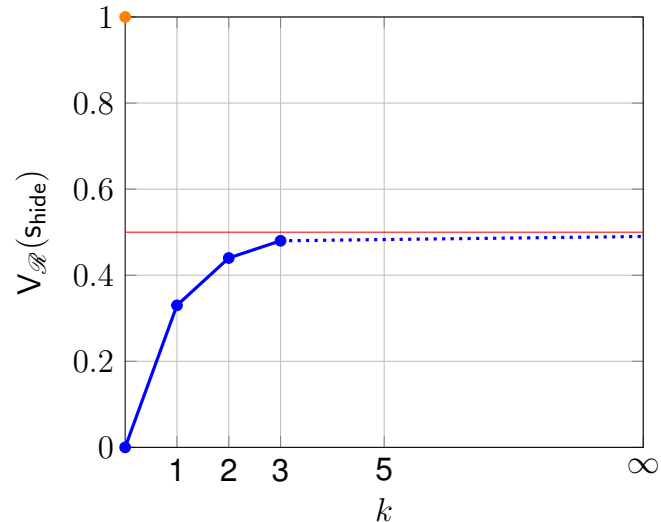
$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$

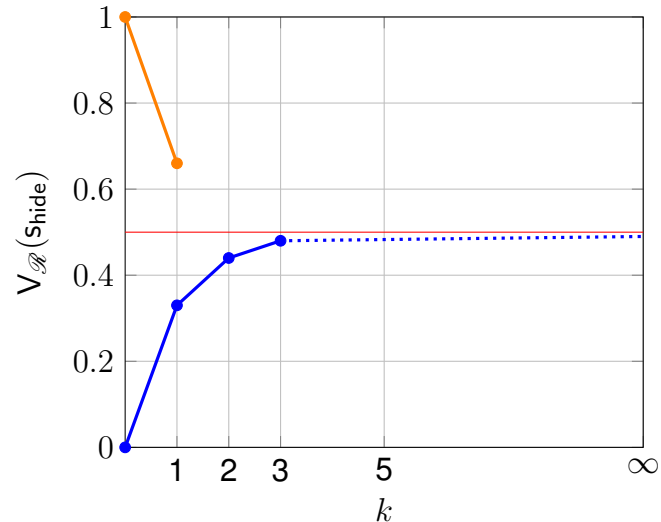
$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$

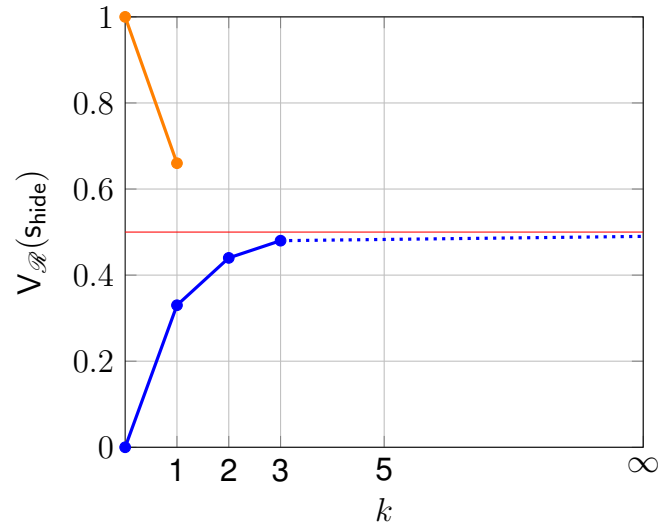
$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot 1 + \frac{1}{3} \\ 1 & 1 \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$

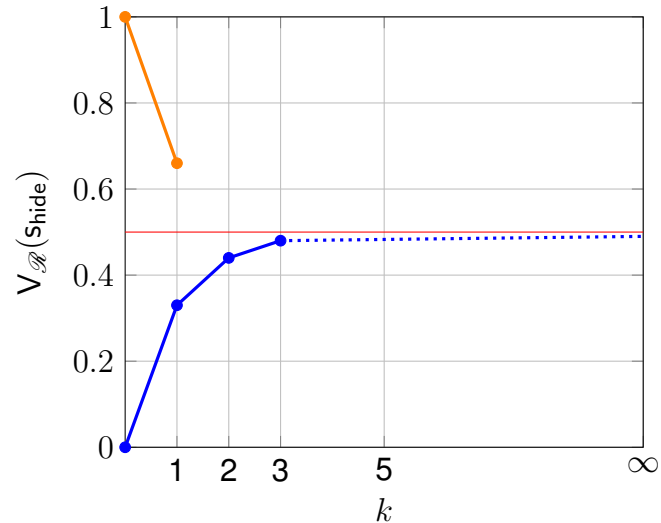
$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{3} \\ 1 & \frac{2}{3} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$

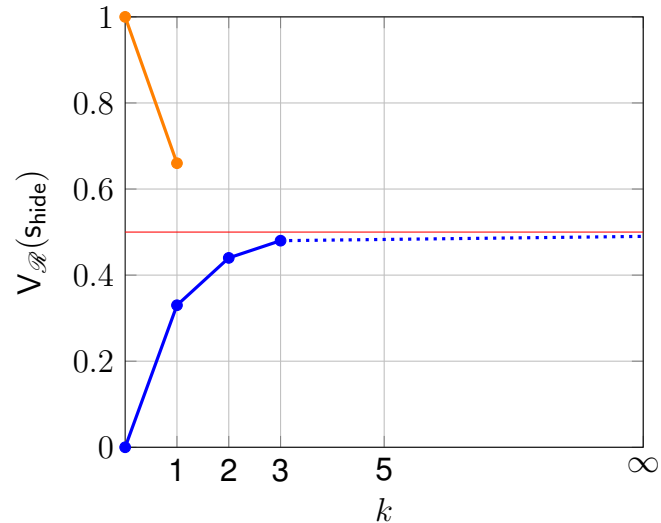
$$\begin{array}{cc}
 \text{throw} & \text{wait} \\
 \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{3} \\ 1 & \frac{2}{3} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array}
 \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$
- Next, deflate $U(s_{\text{hide}})$ to $\frac{5}{9}$

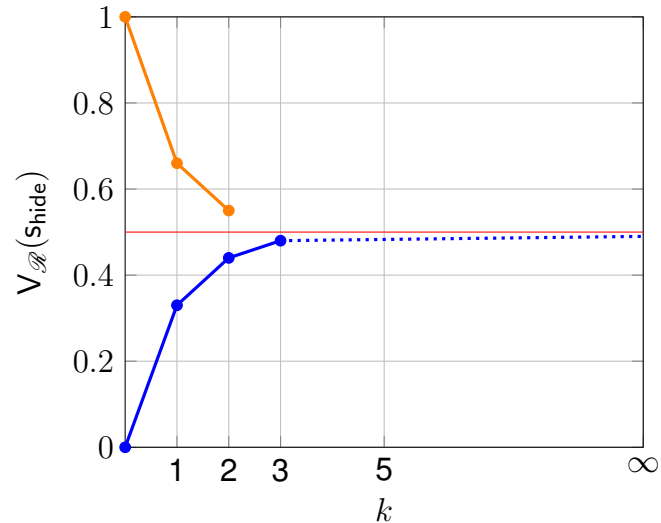
$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{3} \\ 1 & \frac{2}{3} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$
- Next, deflate $U(s_{\text{hide}})$ to $\frac{5}{9}$

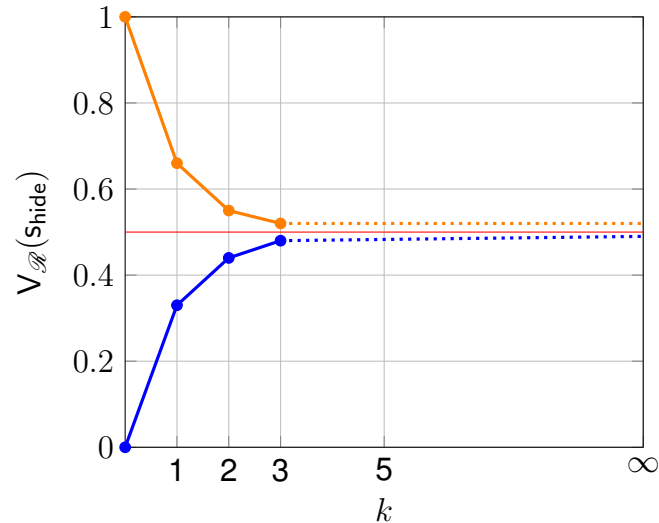
$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{3} \\ 1 & \frac{2}{3} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



Value of Sub-matrix Game Yields Exit Value

- Need to avoid hazardous strategies
- $\{\text{hide} \mapsto 1\}$ is only staying for $\{\text{wait} \mapsto 1\}$
- **Exit value:** Maximal value attainable upon leaving under $\{\text{wait} \mapsto 1\}$
- So, deflate $U(s_{\text{hide}})$ to $\frac{2}{3}$
- Next, deflate $U(s_{\text{hide}})$ to $\frac{5}{9}$
- ...

$$\begin{array}{cc} \text{throw} & \text{wait} \\ \left(\begin{array}{cc} 0 & \frac{1}{3} \cdot \frac{2}{3} + \frac{1}{3} \\ 1 & \frac{2}{3} \end{array} \right) & \begin{array}{l} \text{run} \\ \text{hide} \end{array} \end{array}$$



In General Deflating is Harder

In General Deflating is Harder

- Finding correct sub-matrix game is more involved

$$\begin{array}{ccccc} & d & e & f & \\ \left(\begin{array}{ccc} 0.3 & 0.2 & 0.1 \\ 1 & U(s) & U(s) \\ U(s) & U(s) & 1 \end{array} \right) & a & b & c \end{array}$$

In General Deflating is Harder

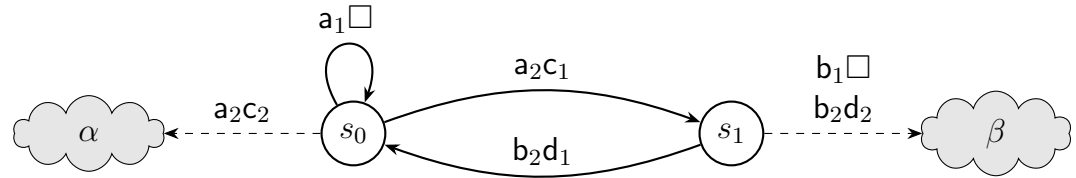
- Finding correct sub-matrix game is more involved
- For more states:

$$\begin{array}{ccccc} & d & e & f & \\ \left(\begin{array}{ccc} 0.3 & 0.2 & 0.1 \\ 1 & U(s) & U(s) \\ U(s) & U(s) & 1 \end{array} \right) & a & b & c \end{array}$$

In General Deflating is Harder

- Finding correct sub-matrix game is more involved
- For more states:
 - Need to find the *best* exit value

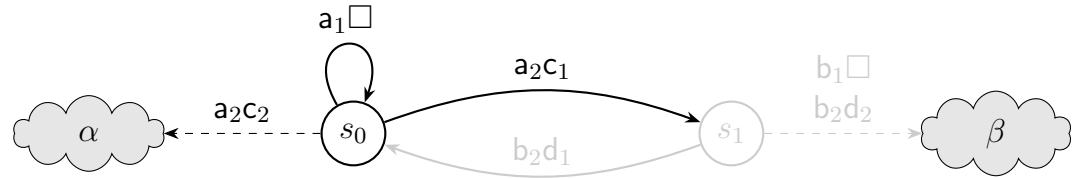
$$\begin{array}{ccc|c}
 & d & e & f \\
 \begin{pmatrix} 0.3 & 0.2 & 0.1 \\ 1 & U(s) & U(s) \\ U(s) & U(s) & 1 \end{pmatrix} & a & b & c
 \end{array}$$



In General Deflating is Harder

- Finding correct sub-matrix game is more involved
- For more states:
 - Need to find the *best* exit value
 - Recursive handling might be necessary

$$\begin{array}{ccc|c}
 d & e & f & \\
 \hline
 \begin{pmatrix} 0.3 & 0.2 & 0.1 \\ 1 & U(s) & U(s) \\ U(s) & U(s) & 1 \end{pmatrix} & a & b & c
 \end{array}$$



Summary

Summary

- Better understanding for CSGs

Summary

- Better understanding for CSGs
 - Optimal strategies are dominant

Summary

- Better understanding for CSGs
 - Optimal strategies are dominant
 - Hazardous strategies hinder convergence

Summary

- Better understanding for CSGs
 - Optimal strategies are dominant
 - Hazardous strategies hinder convergence
 - Exiting sub-game is needed for deflating

Summary

- Better understanding for CSGs
 - Optimal strategies are dominant
 - Hazardous strategies hinder convergence
 - Exiting sub-game is needed for deflating
- Now: Stopping criterion for bounded VI in CSGs

Summary

- Better understanding for CSGs
 - Optimal strategies are dominant
 - Hazardous strategies hinder convergence
 - Exiting sub-game is needed for deflating
- Now: Stopping criterion for bounded VI in CSGs

Thank you!