

Teaching our PET: Should I Stay or Should I Go?

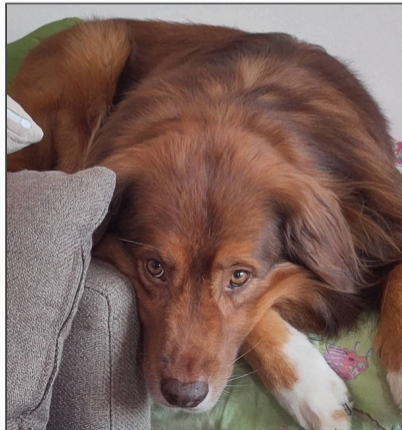
Extending the Partial Exploration Tool to Stochastic Games

Tobias Meggendorfer Maximilian Weininger[†]

[†]Supported by Marie Skłodowska-Curie grant No. 101034413



CAV 24



Teaching our PET: Should I Stay or Should I Go?

Extending the Partial Exploration Tool to Stochastic Games

Tobias Meggendorfer Maximilian Weininger[†]

[†]Supported by Marie Skłodowska-Curie grant No. 101034413



CAV 24



```
while (! verdict.isSolved(Bounds.of(lower[initialState], upper[initialState]))) {  
    // Basic value iteration  
    statistics.reachIterationTime.start();  
    var activeStateIterator = activeStates.iterator();  
    while (activeStateIterator.hasNext()) {  
        int state = activeStateIterator.nextInt();  
        if (FloatUtil.isZero(updateBounds(state))) {  
            activeStateIterator.remove();  
        }  
    }  
    statistics.reachIterationTime.stop();  
  
    // Detect new SECs  
    statistics.secDiscoveryTime.start();  
    componentUpdates.performUpdates();  
    statistics.secDiscoveryTime.stop();  
  
    // Deflate  
    statistics.flateCount += components.size();  
    statistics.flateTime.start();  
    for (GameComponent<L> component : components) {  
        component.secs.removeIf(this::deflate);  
    }  
    statistics.flateTime.stop();  
  
    step += 1;  
    statistics.reachIterations += 1;  
    logProgress.step();  
}
```

Teaching our PET: Should I Stay or Should I Go?

Extending the Partial Exploration Tool to Stochastic Games

Tobias Meggendorfer **Maximizer Weininger**[†]

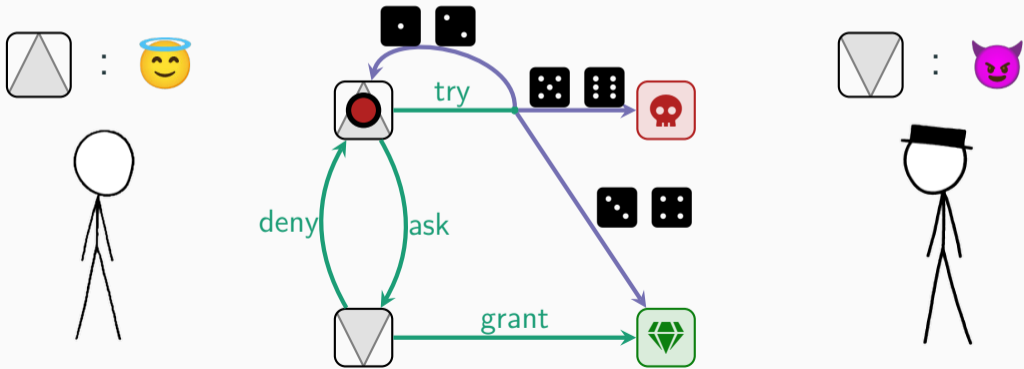
[†]Supported by Marie Skłodowska-Curie grant No. 101034413

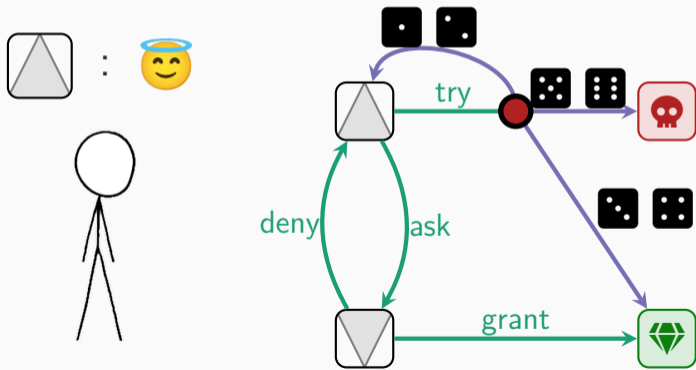


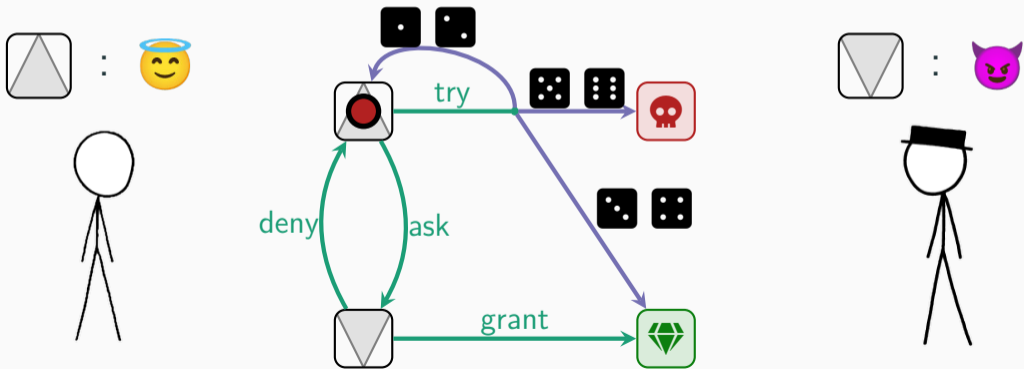
CAV 24

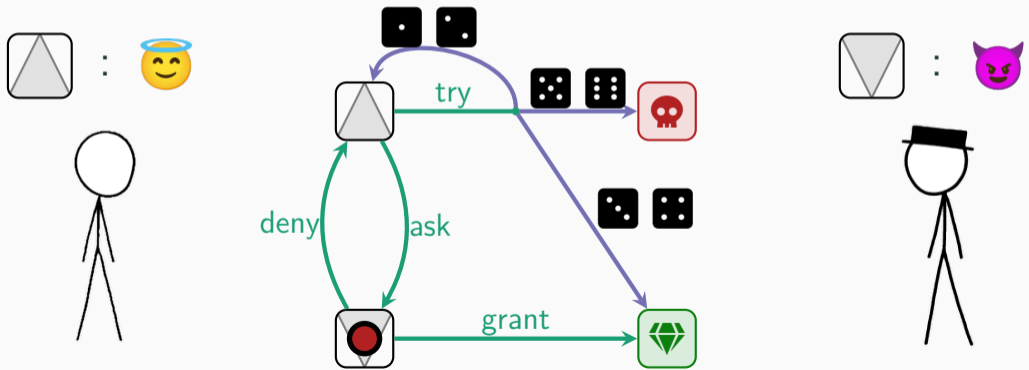


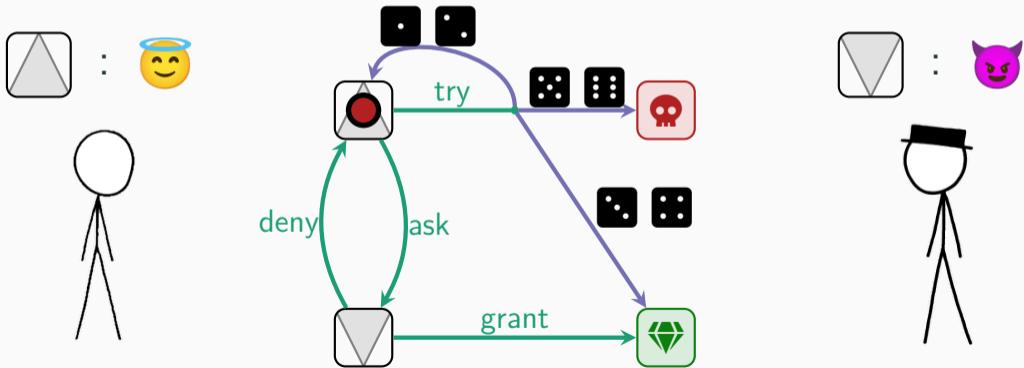
```
while (! verdict.isSolved(Bounds.of(lower[initialState], upper[initialState]))) {  
    // Basic value iteration  
    statistics.reachIterationTime.start();  
    var activeStateIterator = activeStates.iterator();  
    while (activeStateIterator.hasNext()) {  
        int state = activeStateIterator.nextInt();  
        if (FloatUtil.isZero(updateBounds(state))) {  
            activeStateIterator.remove();  
        }  
    }  
    statistics.reachIterationTime.stop();  
  
    // Detect new SECs  
    statistics.secDiscoveryTime.start();  
    componentUpdates.performUpdates();  
    statistics.secDiscoveryTime.stop();  
  
    // Deflate  
    statistics.flateCount += components.size();  
    statistics.flateTime.start();  
    for (GameComponent<L> component : components) {  
        component.secs.removeIf(this::deflate);  
    }  
    statistics.flateTime.stop();  
  
    step += 1;  
    statistics.reachIterations += 1;  
    logProgress.step();  
}
```

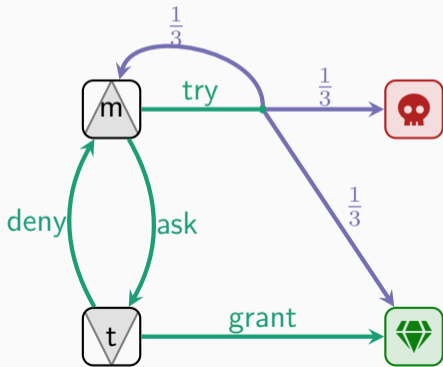




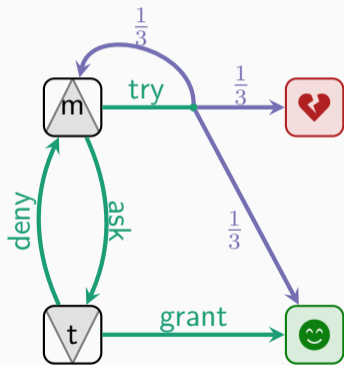




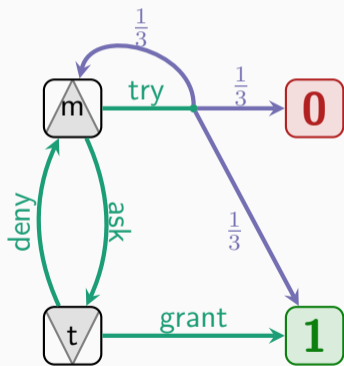




Idea: Value of state = optimum over actions : expected value of successors

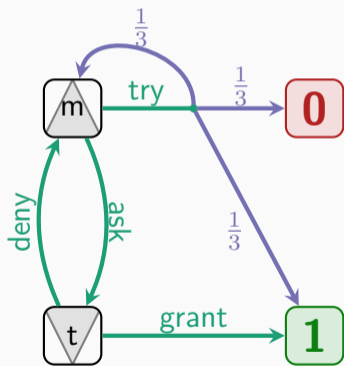


Idea: Value of state = optimum over actions : expected value of successors



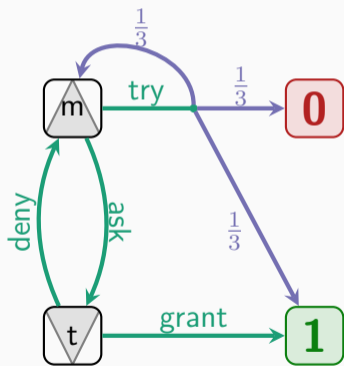
m	0
t	0

Idea: Value of state = optimum over actions : expected value of successors



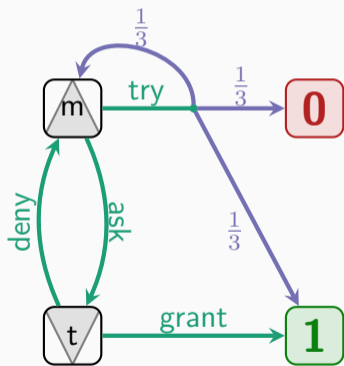
m	0	1/3
t	0	0

Idea: Value of state = optimum over actions : expected value of successors



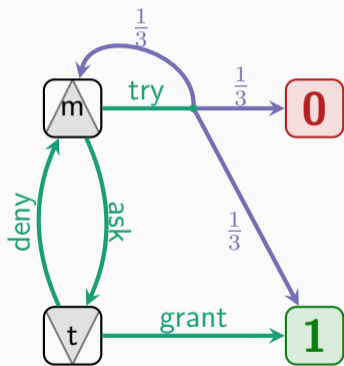
m	0	1/3	4/9
t	0	0	1/3

Idea: Value of state = optimum over actions : expected value of successors



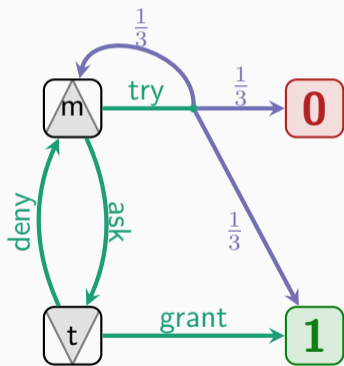
m	0	1/3	4/9	...
t	0	0	1/3	...

Idea: Value of state = optimum over actions : expected value of successors



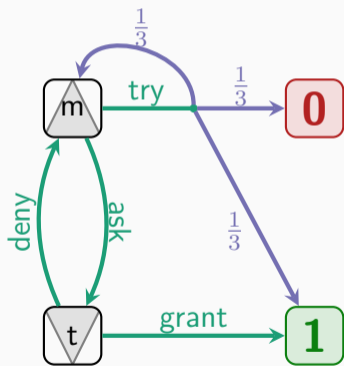
m	0	1/3	4/9	...
t	0	0	1/3	...
m	1			
t	1			

Idea: Value of state = optimum over actions : expected value of successors



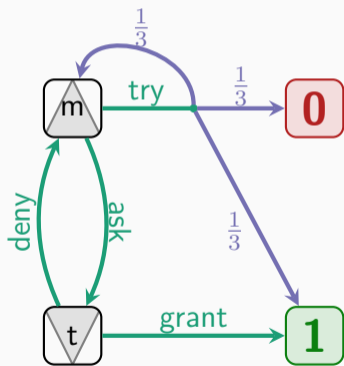
m	0	$\frac{1}{3}$	$\frac{4}{9}$...
t	0	0	$\frac{1}{3}$...
m	1	1		
t	1	1		

Idea: Value of state = optimum over actions : expected value of successors



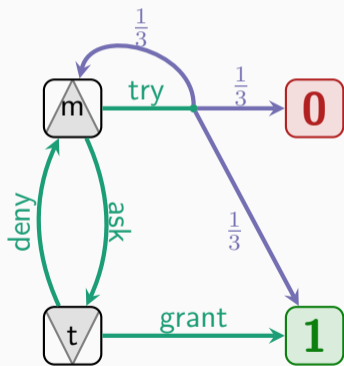
m	0	1/3	4/9	...
t	0	0	1/3	...
m	1	1	1	
t	1	1	1	

Idea: Value of state = optimum over actions : expected value of successors



m	0	$\frac{1}{3}$	$\frac{4}{9}$...
t	0	0	$\frac{1}{3}$...
m	1	1	1	...
t	1	1	1	...

Idea: Value of state = optimum over actions : expected value of successors



m 0 1/3 4/9 ...

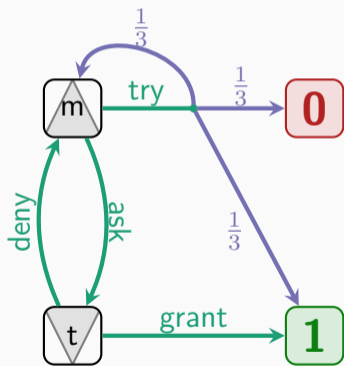
t 0 0 1/3 ...

m 1 1 1 ...

t 1 1 1 ...



Idea: Value of state = optimum over actions : expected value of successors



m 0 1/3 4/9 ...

t 0 0 1/3 ...

m 1 1 1 ...

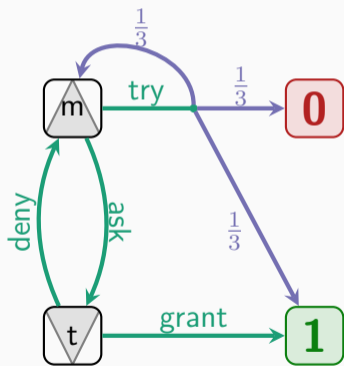
t 1 1 1 ...



m 1

t 1

Idea: Value of state = optimum over actions : expected value of successors



m	0	1/3	4/9	...
---	---	-----	-----	-----

t	0	0	1/3	...
---	---	---	-----	-----

m	1	1	1	...
---	---	---	---	-----

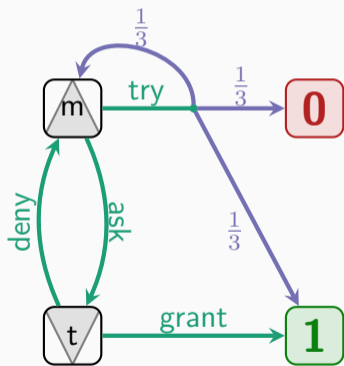
t	1	1	1	...
---	---	---	---	-----



m	1	2/3
---	---	-----

t	1	1
---	---	---

Idea: Value of state = optimum over actions : expected value of successors



m	0	1/3	4/9	...
---	---	-----	-----	-----

t	0	0	1/3	...
---	---	---	-----	-----

m	1	1	1	...
---	---	---	---	-----

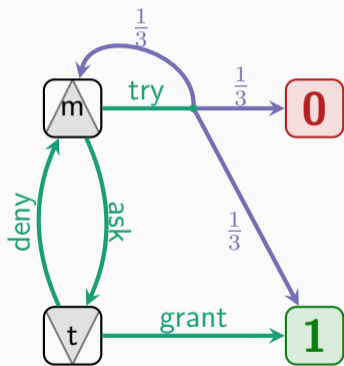
t	1	1	1	...
---	---	---	---	-----



m	1	2/3	5/9	...
---	---	-----	-----	-----

t	1	1	2/3	...
---	---	---	-----	-----

Idea: Value of state = optimum over actions : expected value of successors



m	0	1/3	4/9	...
---	---	-----	-----	-----

t	0	0	1/3	...
---	---	---	-----	-----

m	1	1	1	...
---	---	---	---	-----

t	1	1	1	...
---	---	---	---	-----



m	1	2/3	5/9	...
---	---	-----	-----	-----

t	1	1	2/3	...
---	---	---	-----	-----



Value iteration alone: May get stuck! So ask:

Should I stay or should I go?

Compare staying forever vs. best exit

SOUND

Updates are correct

Value iteration alone: May get stuck! So ask:

Should I stay or should I go?

Compare staying forever vs. best exit

SOUND

Updates are correct

COMPLETE

Updates ensures convergence

Value iteration alone: May get stuck! So ask:

Should I stay or should I go?

Compare staying forever vs. best exit

SOUND

Updates are correct

COMPLETE

Updates ensures convergence

Value iteration alone: May get stuck! So ask:
Should I stay or should I go?
Compare staying forever vs. best exit

UNIVERSAL

(High-level) Reasoning is objective independent

[KMW23]: **Unifies and extends previous works**

[Brá+14; HM14; Bai+17; Ash+17] (MDP) [Kel+18; Pha+20] (SG)

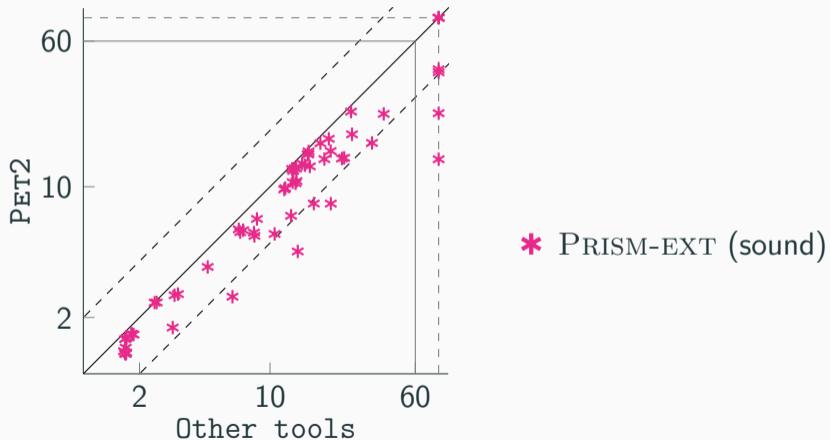
PET1 \rightarrow PET2:

- Partial exploration extended to SG
- Global approach for all supported models

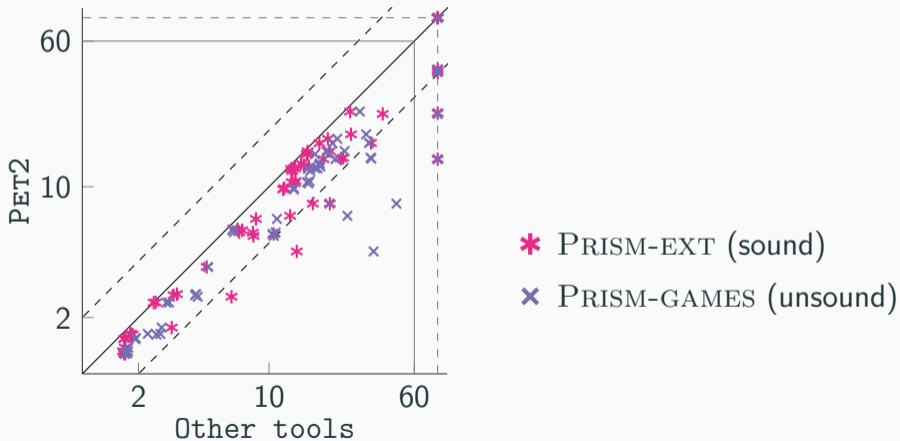
PET1 \rightarrow PET2:

- Partial exploration extended to SG
 - Global approach for all supported models
 - Numerous engineering improvements
- } each with unified algorithm

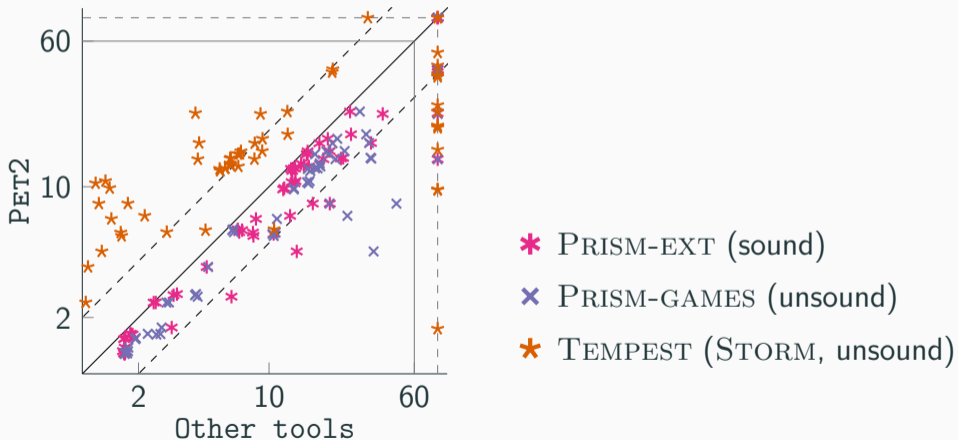
Comparing PET2 (global approach) on stochastic games



Comparing PET2 (global approach) on stochastic games



Comparing PET2 (global approach) on stochastic games



PET: A tool implementing the unified theory for handling stochastic games

Read the tool paper



Tobias Meggendorfer and Maximilian Weininger. “Playing Games with your PET: Extending the Partial Exploration Tool to Stochastic Games.” In: *CAV. 2024*

Your best exits?

Read the theory paper



Jan Křetínský, Tobias Meggendorfer, and Maximilian Weininger. “Stopping Criteria for Value Iteration on Stochastic Games with Quantitative Objectives.” In: *LICS. 2023*

Visit the webpage



(Poster, Talk, Papers, ...)