Practical Introduction to Safe Reinforcement Learning

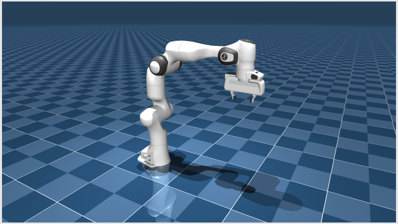
Kryspin Varys

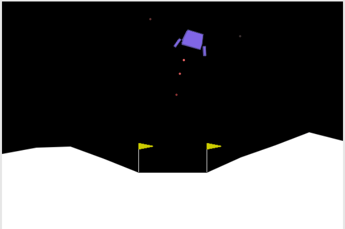
University of Southampton

4 February 2024

Outline

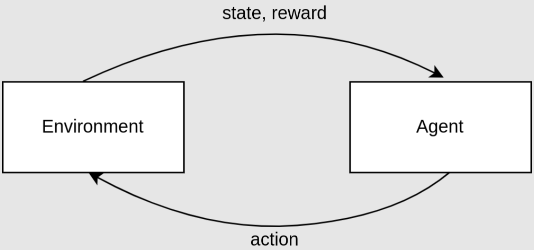
To solve control problems:

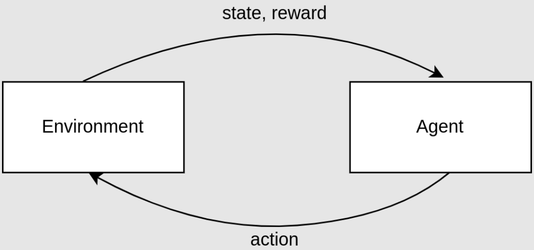




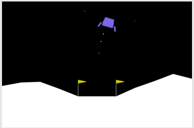
Introduction to Safe RL

What is RL?





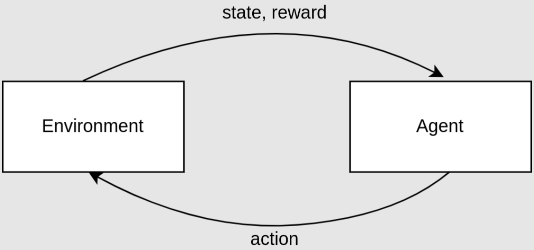
discrete env.



continuous env.

The environment:

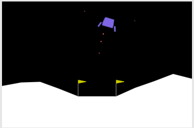
$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, \quad R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, \quad T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$



discrete env.

The agent π is either:

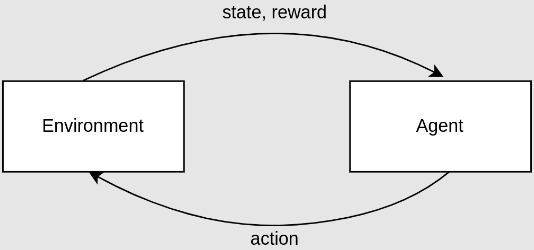
1. table in the case of small discrete spaces, or
2. neural network in the case of large spaces.



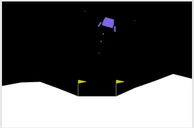
continuous env.

Introduction to Safe RL

What is RL?



discrete env.



continuous env.

The agent π aims to maximize an optimality criterion:

$$\max \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \right]$$

The agent's lifecycle has two phases:

1. the traning phase and
2. the deployment phase.

Training

1. Exploration
   1.1 Random actions
2. Exploitation
   2.1 Best actions according to the optimality criterion

Training

1. Exploration
   1.1 Random actions
2. Exploitation
   2.1 Best actions according to the optimality criterion

Deployment

1. Exploitation

Training

1. Exploration
   1.1 Random actions
2. Exploitation
   2.1 Best actions according to the optimality criterion

Deployment

1. Exploitation

The agent π aims to maximize the optimality criterion:

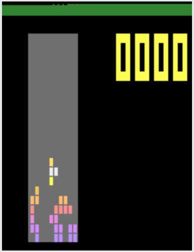
$$\max \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t)\right]$$

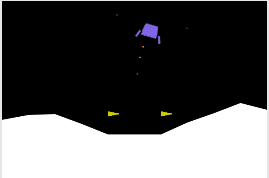
Reinforcement learning is enabled by many great projects such as:

[1] `https://gymnasium.farama.org/`

Reinforcement learning is enabled by many great projects such as:

1. OpenAI's Gymnasium [1],



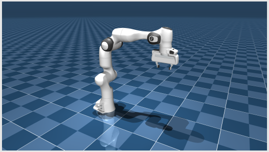
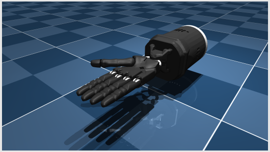
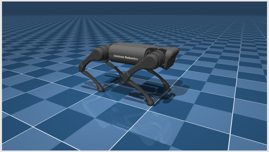


[1] https://gymnasium.farama.org/

Introduction to Safe RL
Open-Source Projects

Reinforcement learning is enabled by many great projects such as:

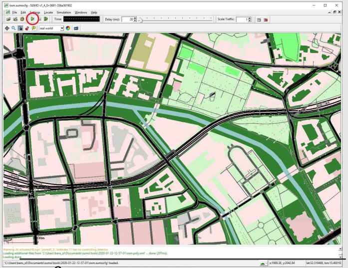
1. OpenAI's Gymnasium,
2. DeepMind's MuJoCo [2],

[2] https://mujoco.org/

Reinforcement learning is enabled by many great projects such as:

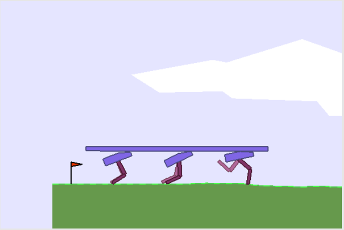
1. OpenAI's Gymnasium,
2. DeepMind's MuJoCo,
3. SUMO and Carla [3].

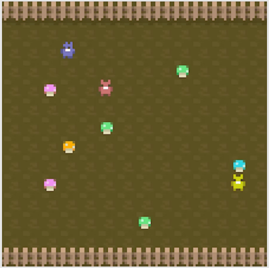


[3] https://eclipse.dev/sumo/ and http://carla.org/

Reinforcement learning is enabled by many great projects such as:

1. OpenAI's Gymnasium,
2. DeepMind's MuJoCo,
3. SUMO and Carla,
4. PettingZoo and Melting Pot [4].





[4] pettingzoo.farama.org/ github.com/google-deepmind/meltingpot

```
import gymnasium as gym
class YourEnv(gym.env):
    ⋮
    def step(action) -> reward, state:
        # environmnent dynamics
        ⋮
        return reward, state
```

Introduction to Safe RL

OpenAI API (Simplified)

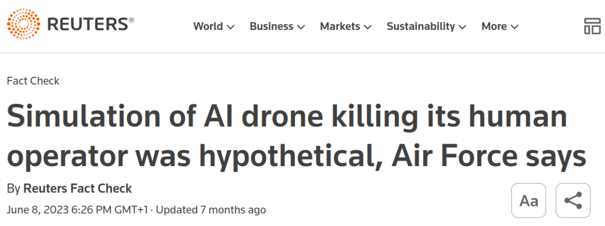
```
import gymnasium as gym
class YourEnv(gym.env):
    ⋮
    def step(action) -> reward, state:
        # environmnent dynamics
        ⋮
        return reward, state

class Agent:
    ⋮
    def act(reward, state) -> action:
        # agent dynamics
        ⋮
        return action
```

Introduction to Safe RL

Why Safe RL?

Agent trying to maximize an optimality criterion must be creative.

This creativity has the potential to endanger the agent or its environment [5].


REUTERS®

World Business Markets Sustainability More

Fact Check

Simulation of AI drone killing its human operator was hypothetical, Air Force says

By **Reuters Fact Check**

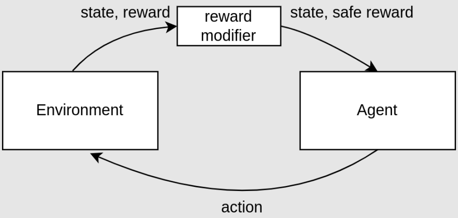
June 8, 2023 6:26 PM GMT+1 · Updated 7 months ago

Aa

[1] https://www.reuters.com/article/factcheck-ai-drone-kills-idUSL1N38023R/

Introduction to Safe RL

What makes RL safe?[6]

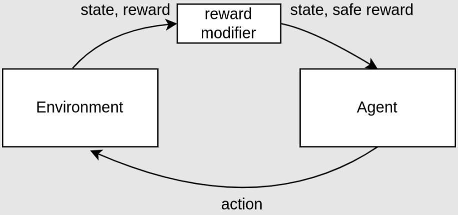


Modifying the optimality criterion to include safety.

[6] Javier Garcia and Fernando Fernandez. "A Comprehensive Survey on Safe Reinforcement Learning". In: *J. Mach. Learn. Res.* 16.1 (Jan. 2015). ISSN: 1532-4435.

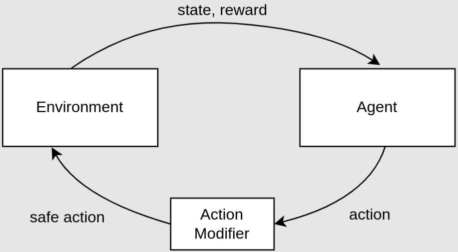
Introduction to Safe RL

What makes RL safe?[6]



Modifying the optimality criterion to include safety.

Modifying the agent's actions to ensure safety.



[6] Javier Garcia and Fernando Fernandez. "A Comprehensive Survey on Safe Reinforcement Learning". In: *J. Mach. Learn. Res.* 16.1 (Jan. 2015). ISSN: 1532-4435.

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

[7] Yueh-Hua Wu and Shou-De Lin. "A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents". In: *Proceedings of the Thirty-Second AAAI Conference on AI*. AAAI'18/IAAI'18/EAAI'18. 2018.

Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

To make the agent consider safety, we modify the original reward function R:

$$\hat{R} = R + H$$

where $H : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is our safety modification.

[7] Yueh-Hua Wu and Shou-De Lin. "A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents". In: *Proceedings of the Thirty-Second AAAI Conference on AI*. AAAI'18/IAAI'18/EAAI'18. 2018.

Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

To make the agent consider safety, we modify the original reward function R:

$$\hat{R} = R + H$$

where $H : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is our safety modification.

How to obtain H:

1. self-engineer it,
2. infer from some data[7].

[7] Yueh-Hua Wu and Shou-De Lin. "A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents". In: *Proceedings of the Thirty-Second AAAI Conference on AI*. AAAI'18/IAAI'18/EAAI'18. 2018.

Practical Scenarios

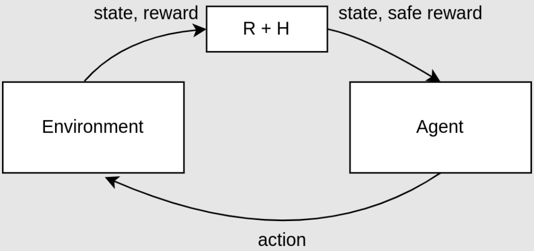
Scenario 1: Modification of the Optimality Criterion

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$\max \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(R(s_t, a_t) + H(s_t, a_t) \right) \right]$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



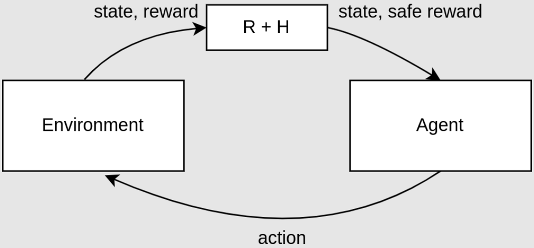
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 1:

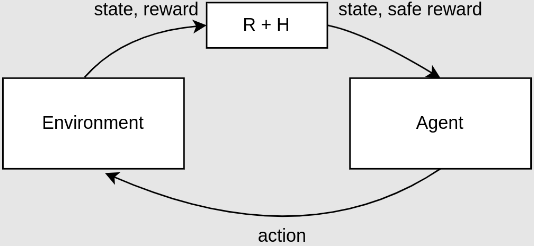
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



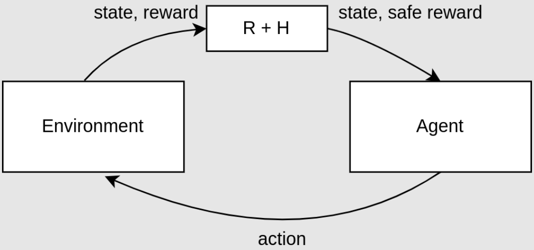
Trajectory 1:

1. reward is -1 + 0

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 1:

1. reward is -1 + 0
2. reward is -1 + 0

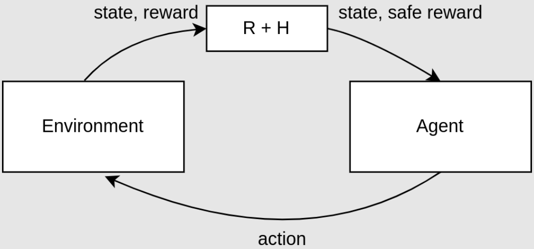
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 1:

1. reward is -1 + 0
2. reward is -1 + 0
3. reward is -1 + 0

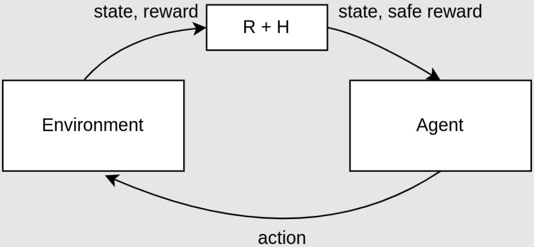
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 1:

1. reward is -1 + 0
2. reward is -1 + 0
3. reward is -1 + 0
4. reward is -1 + 0

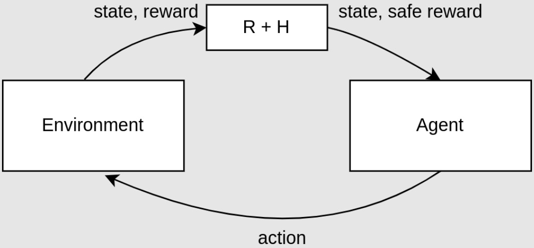
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 1:

1. reward is -1 + 0
2. reward is -1 + 0
3. reward is -1 + 0
4. reward is -1 + 0
5. reward is -1 + 0

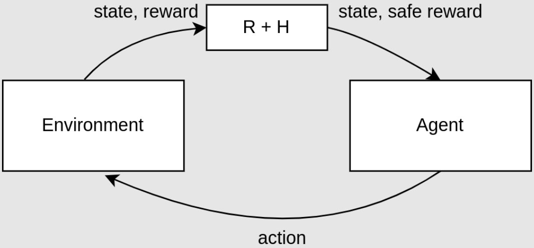
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



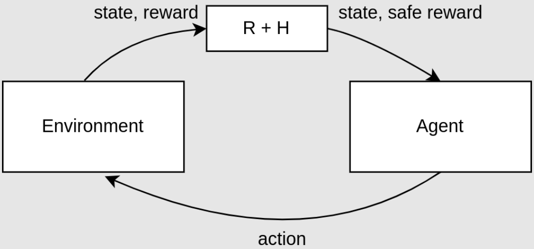
Trajectory 1:

1. reward is -1 $+$ 0
2. reward is -1 $+$ 0
3. reward is -1 $+$ 0
4. reward is -1 $+$ 0
5. reward is -1 $+$ 0
6. reward is 100 $+$ 0

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 2:

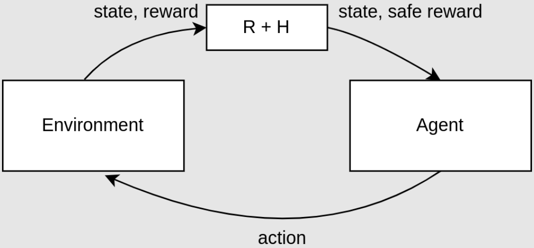
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



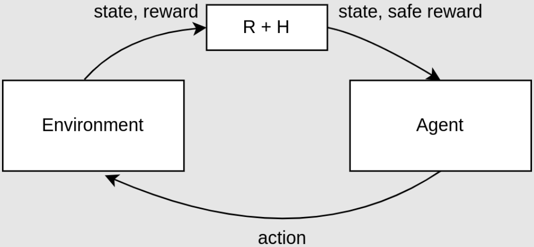
Trajectory 2:

1. reward is -1 + 0

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 2:

1. reward is -1 $+$ 0
2. reward is -1 - 100

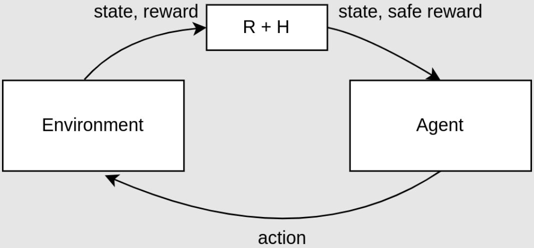
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



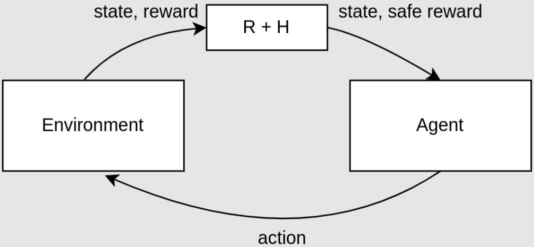
Trajectory 2:

1. reward is -1 + 0
2. reward is -1 - 100
3. reward is -1 + 0

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 2:

1. reward is -1 + 0
2. reward is -1 - 100
3. reward is -1 + 0
4. reward is -1 + 0

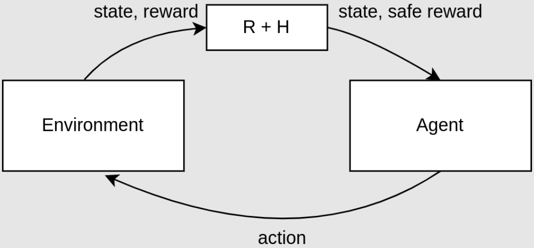
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 2:

1. reward is -1 + 0
2. reward is -1 - 100
3. reward is -1 + 0
4. reward is -1 + 0
5. reward is -1 + 0

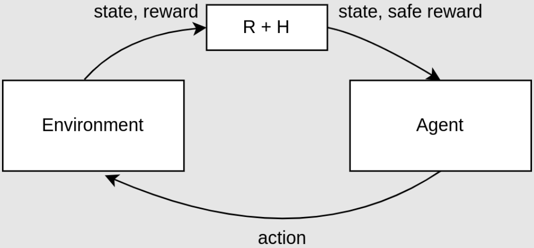
Practical Scenarios

Scenario 1: Modification of the Optimality Criterion

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

$$H = \begin{cases} -100 & \text{reaching water} \\ 0 & \text{otherwise} \end{cases}$$



Trajectory 2:

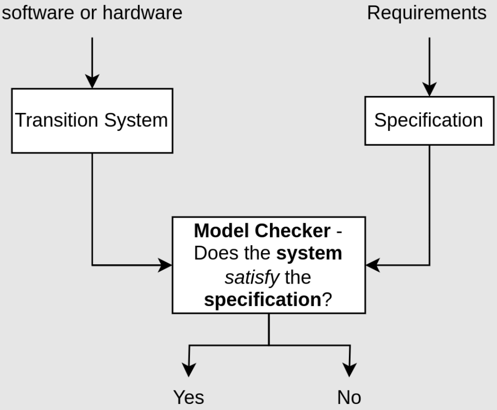
1. reward is -1 + 0
2. reward is -1 - 100
3. reward is -1 + 0
4. reward is -1 + 0
5. reward is -1 + 0
6. reward is 100 + 0

Properties:

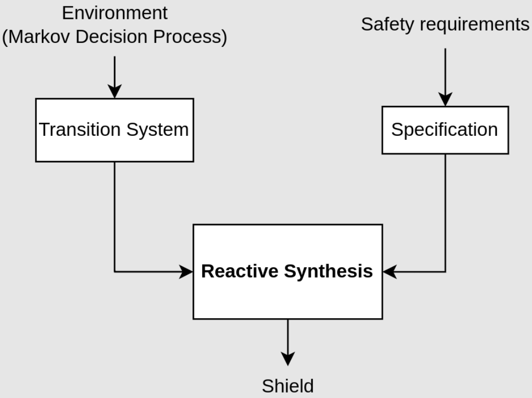
1. Safety only during the deployment phase.
2. Requires the dataset of safe behaviours.
3. We don't need to define what "safety" means.

Scenario 2: Modification of the Agent's Actions

Formal methods:



Formal methods for reinforcement learning[8]:

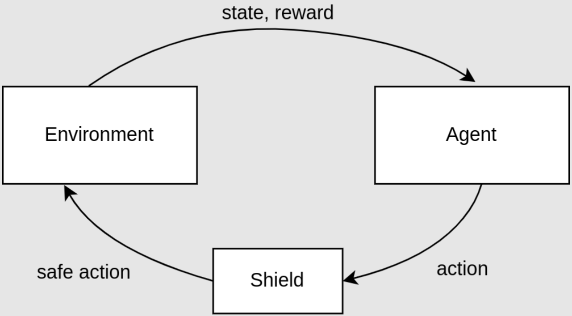


[8] Mohammed Alshiekh et al. "Safe Reinforcement Learning via Shielding". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018).

Practical Scenarios

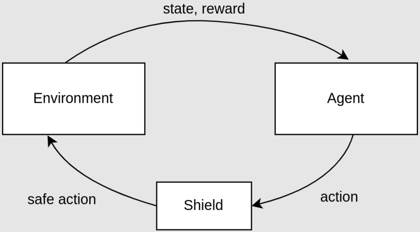
Scenario 2: Modification of the Agent's Actions

Formal methods for reinforcement learning



Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

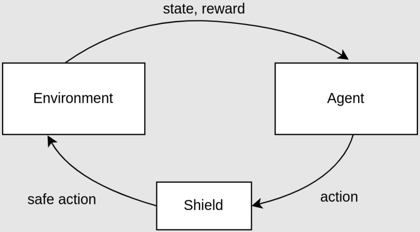
$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 1:

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

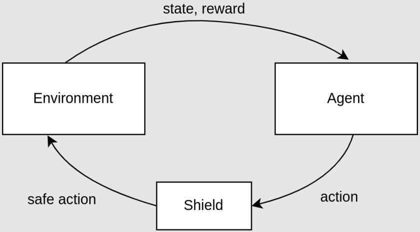


Trajectory 1:

1. reward is -1

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 1:

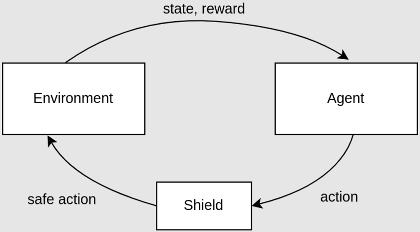
1. reward is -1
2. reward is -1

Practical Scenarios

Scenario 2: Modification of the Agent's Actions

Markov decision process $= \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 1:

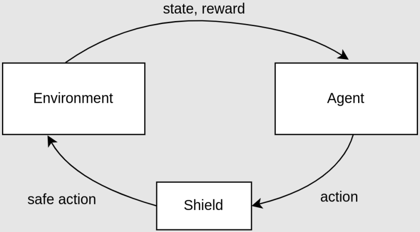
1. reward is -1
2. reward is -1
3. reward is -1

Practical Scenarios

Scenario 2: Modification of the Agent's Actions

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 1:

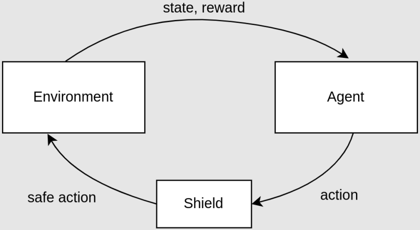
1. reward is -1
2. reward is -1
3. reward is -1
4. reward is -1

Practical Scenarios

Scenario 2: Modification of the Agent's Actions

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 1:

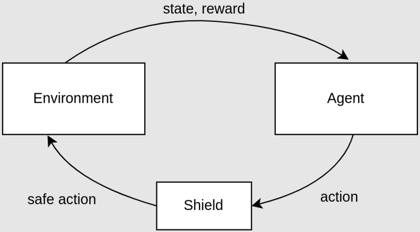
1. reward is -1
2. reward is -1
3. reward is -1
4. reward is -1
5. reward is -1

Practical Scenarios

Scenario 2: Modification of the Agent's Actions

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$

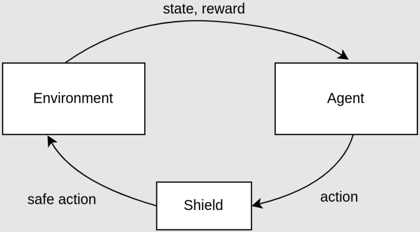


Trajectory 1:

1. reward is -1
2. reward is -1
3. reward is -1
4. reward is -1
5. reward is -1
6. reward is 100

$$\text{Markov decision process} = \langle\, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \,\rangle$$

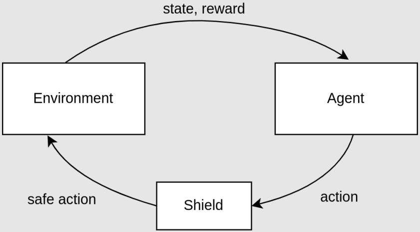
$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 2:

Markov decision process $= \langle \, \mathcal{S}, \mathcal{A}, R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}, T : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \, \rangle$

$$R = \begin{cases} 100 & \text{reaching goal} \\ -1 & \text{otherwise} \end{cases}$$



Trajectory 2:

1. reward is -1

Properties:

1. Keeps the agent provably safe *during training and deployment*.

Properties:

1. Keeps the agent provably safe *during training and deployment*.
2. The guarantee is only with respect to the *transition system*!
3. We must be able to come up with the transition system.
4. We must know the safety specifications.

Thank You!

References I

[1] Mohammed Alshiekh et al. "Safe Reinforcement Learning via Shielding". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (Apr. 2018). URL: https://ojs.aaai.org/index.php/AAAI/article/view/11797.

[2] Javier Garcia and Fernando Fernandez. "A Comprehensive Survey on Safe Reinforcement Learning". In: *J. Mach. Learn. Res.* 16.1 (Jan. 2015), pp. 1437–1480. ISSN: 1532-4435.

[3] Yueh-Hua Wu and Shou-De Lin. "A Low-Cost Ethics Shaping Approach for Designing Reinforcement Learning Agents". In: *Proceedings of the Thirty-Second AAAI Conference on AI*. AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN: 978-1-57735-800-8.