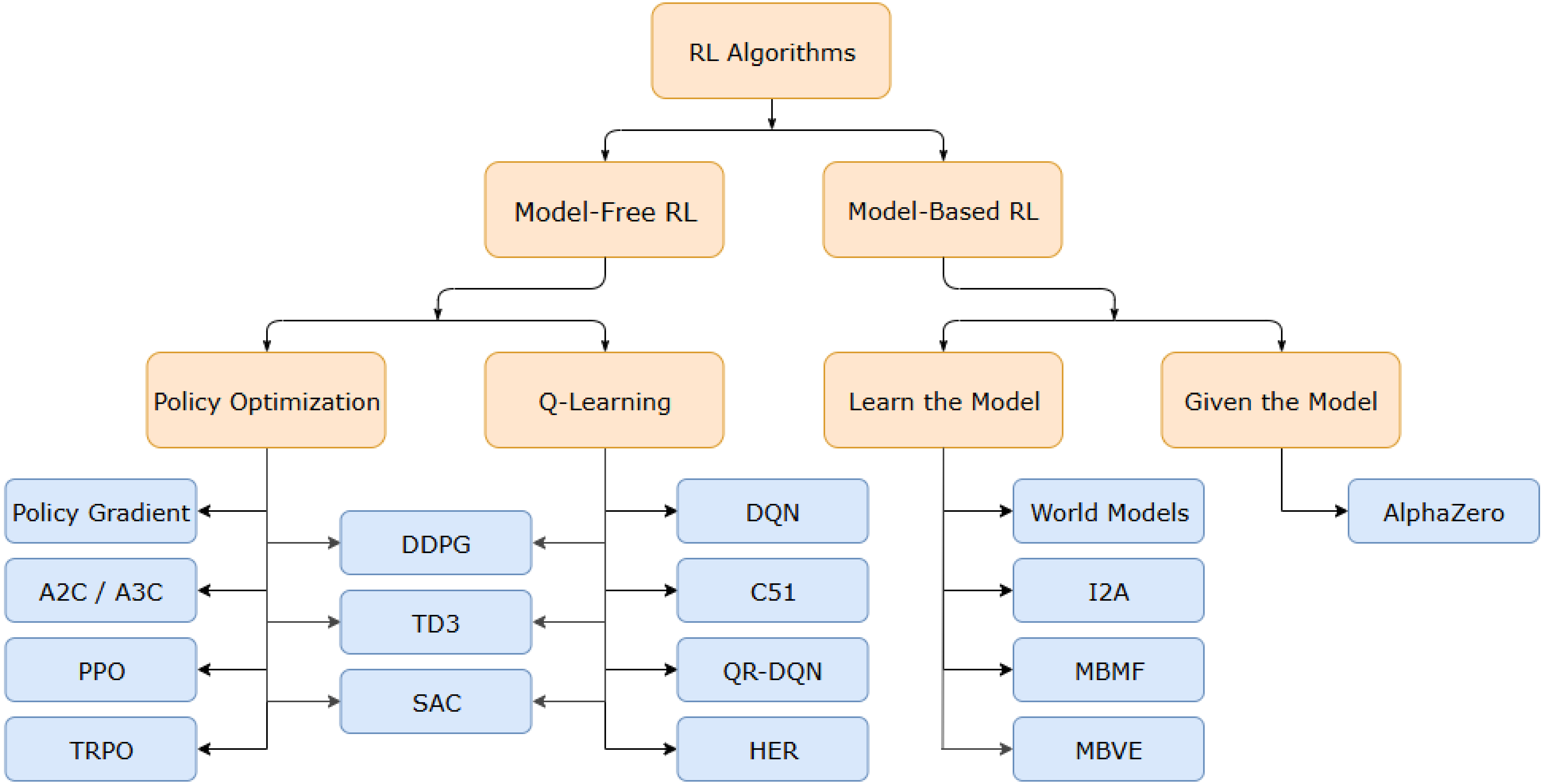
# INTRO
# REINFORCEMENT LEARNING
# (RL)

– Matee Vadrukchid –

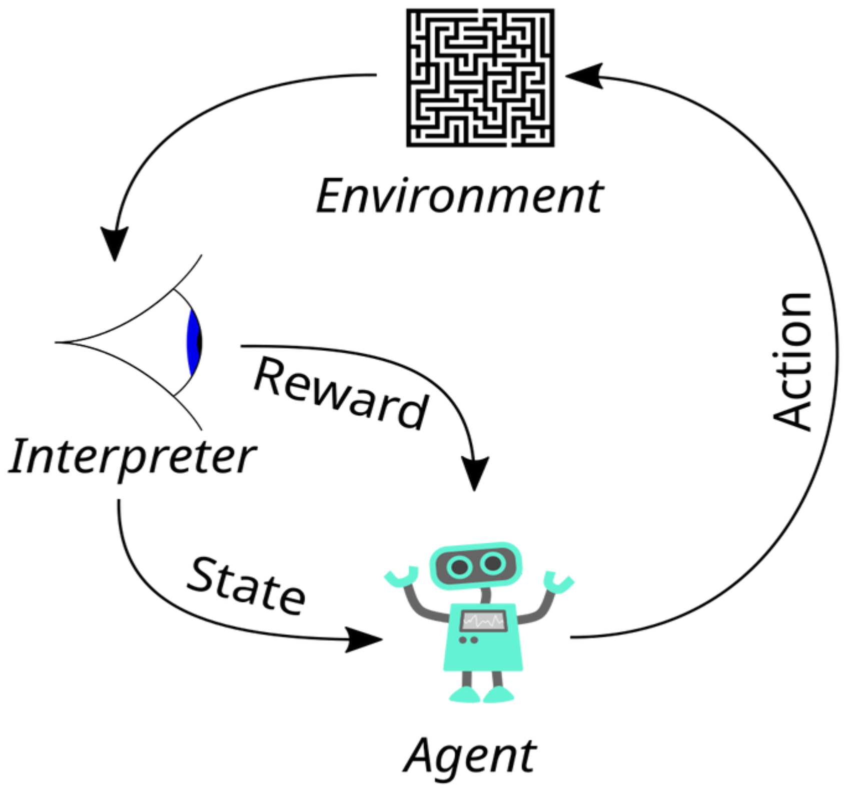
# Big Picture of RL



# Introduction and Outline

## What is Reinforcement Learning?

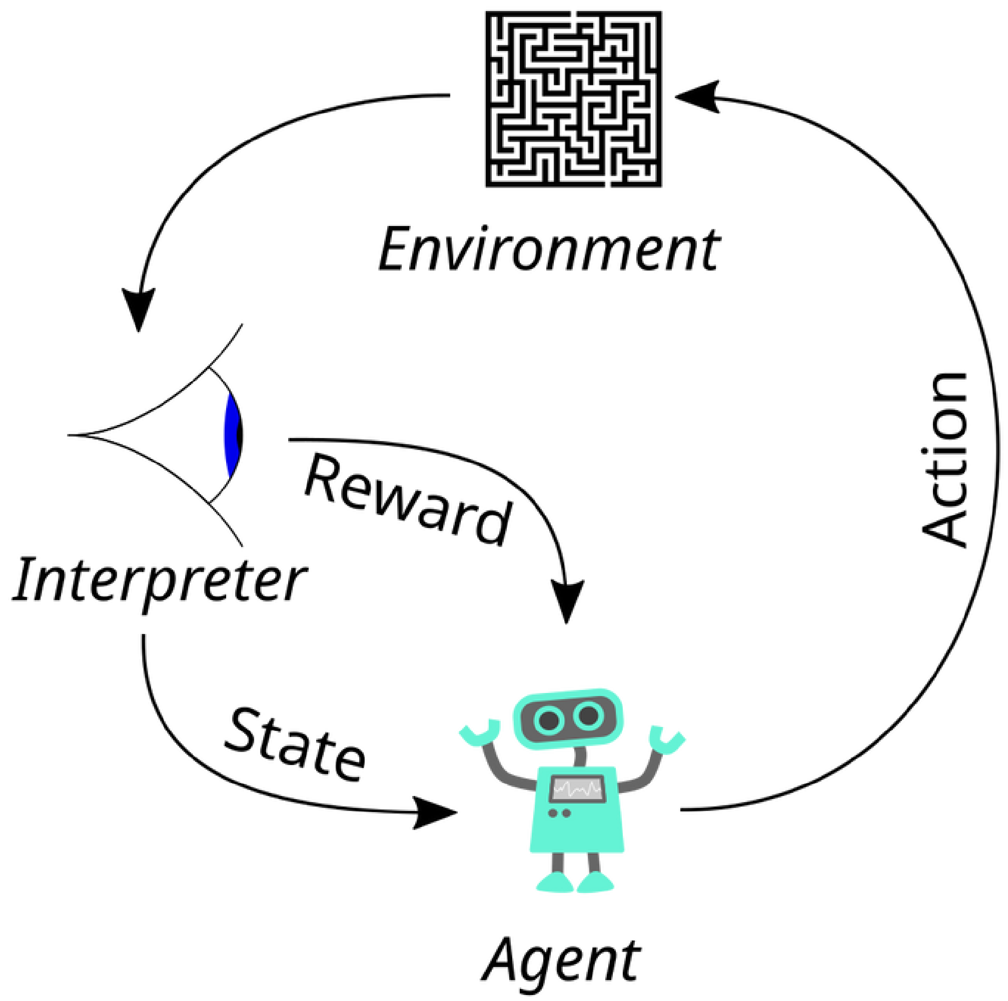
- In Reinforcement Learning, an agent interacts with an environment to learn how to perform a particular task well



Link:https://en.wikipedia.org/wiki/Reinforcement_learning#/media/File:Reinforcement_learning_diagram.svg

# Introduction and Outline

## What is Reinforcement Learning?

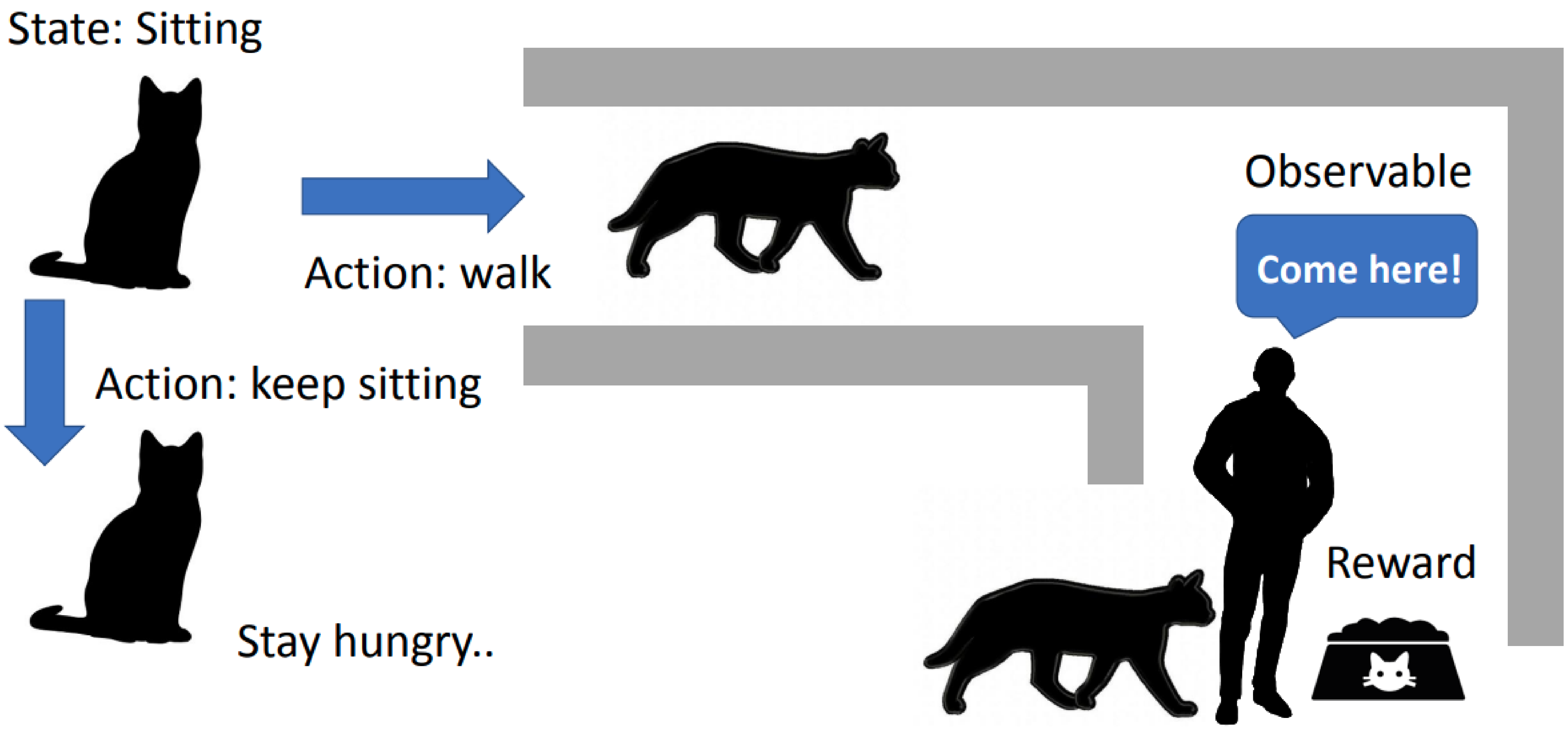


- How is it different to the other learning paradigms?
  - There is no supervisor, only a reward.
  - The agent’s actions affect the subsequent data it receives
  - Feedback is delayed, and may be received after several actions

Link:https://en.wikipedia.org/wiki/Reinforcement_learning#/media/File:Reinforcement_learning_diagram.svg

# Introduction and Outline

## Cat Agent



State: Sitting

Action: walk

Action: keep sitting

Stay hungry..

Observable

Come here!

Reward

# Introduction and Outline

## Examples of Reinforcement Learning

Fly a helicopter

Make a robot walk

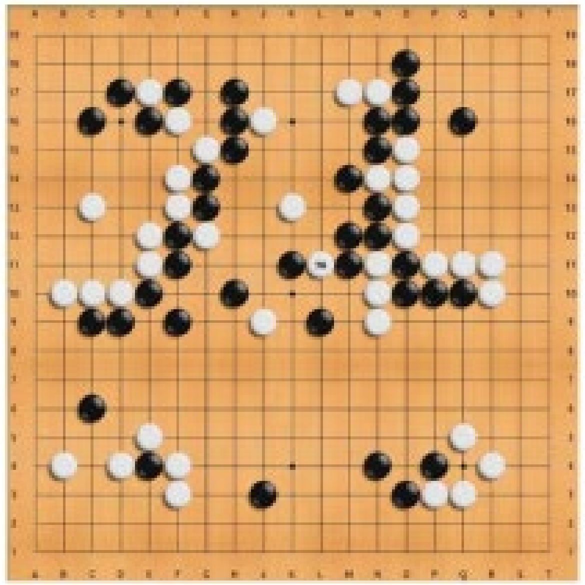
Manage an investment portfolio

Play Atari games better than humans

# Introduction and Outline

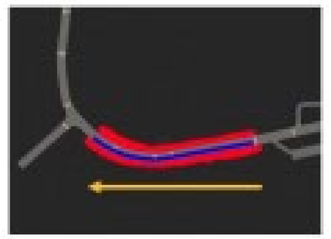
## Example of games

- Chess

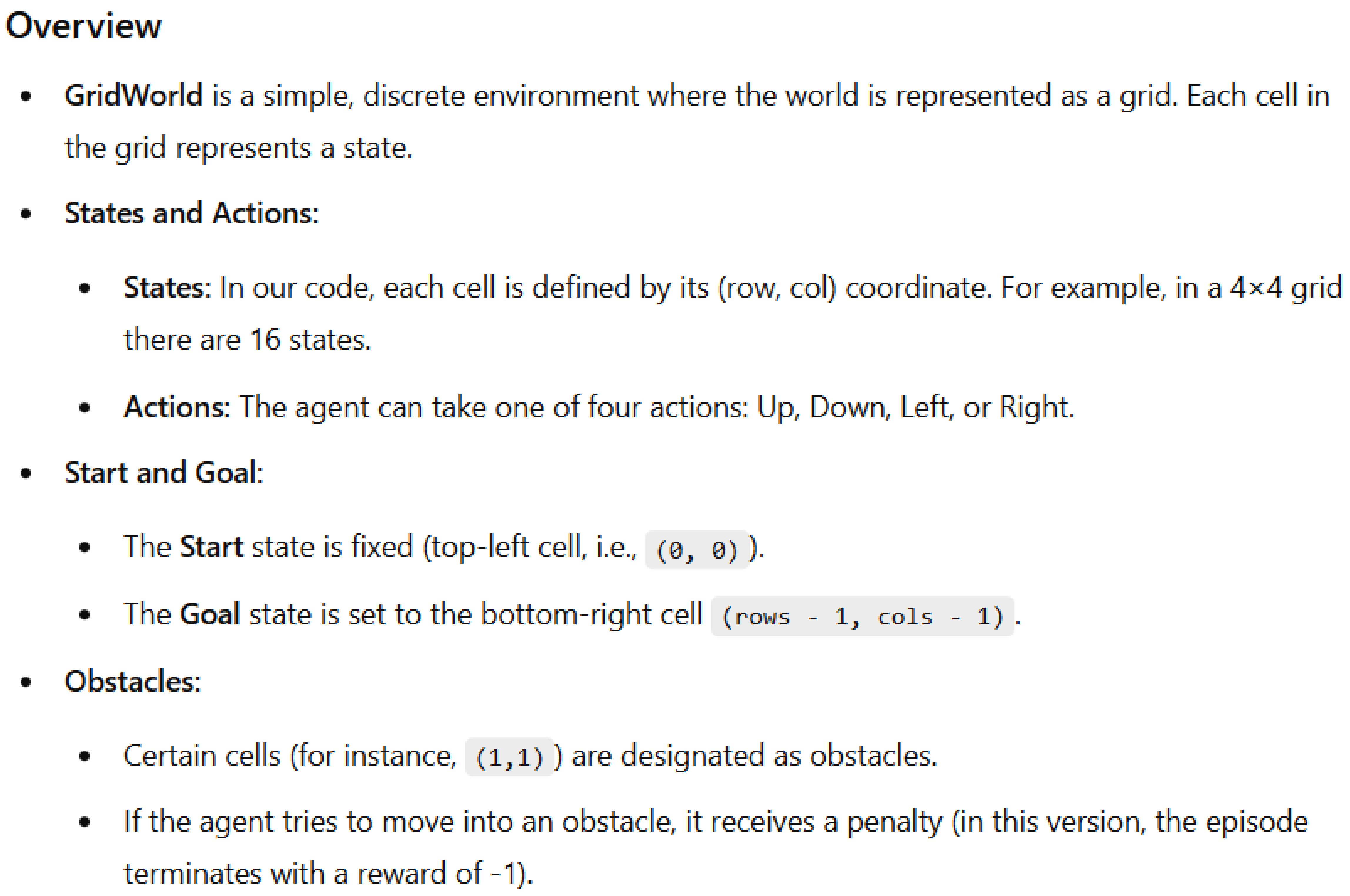


- Go

- Carla



# Explanation & Theory



### Overview

- **GridWorld** is a simple, discrete environment where the world is represented as a grid. Each cell in the grid represents a state.
- **States and Actions:**
  - **States:** In our code, each cell is defined by its (row, col) coordinate. For example, in a 4×4 grid there are 16 states.
  - **Actions:** The agent can take one of four actions: Up, Down, Left, or Right.
- **Start and Goal:**
  - The **Start** state is fixed (top-left cell, i.e., `(0, 0)`).
  - The **Goal** state is set to the bottom-right cell `(rows - 1, cols - 1)`.
- **Obstacles:**
  - Certain cells (for instance, `(1,1)`) are designated as obstacles.
  - If the agent tries to move into an obstacle, it receives a penalty (in this version, the episode terminates with a reward of -1).

# Explanation & Theory

## Out-of-Bound Actions

- If the agent attempts to move off the grid (e.g., moving up from the top row), the environment immediately gives a penalty of -1 and terminates the episode.
- This teaches the agent that certain actions are “illegal” and should be avoided.

## Reward Structure

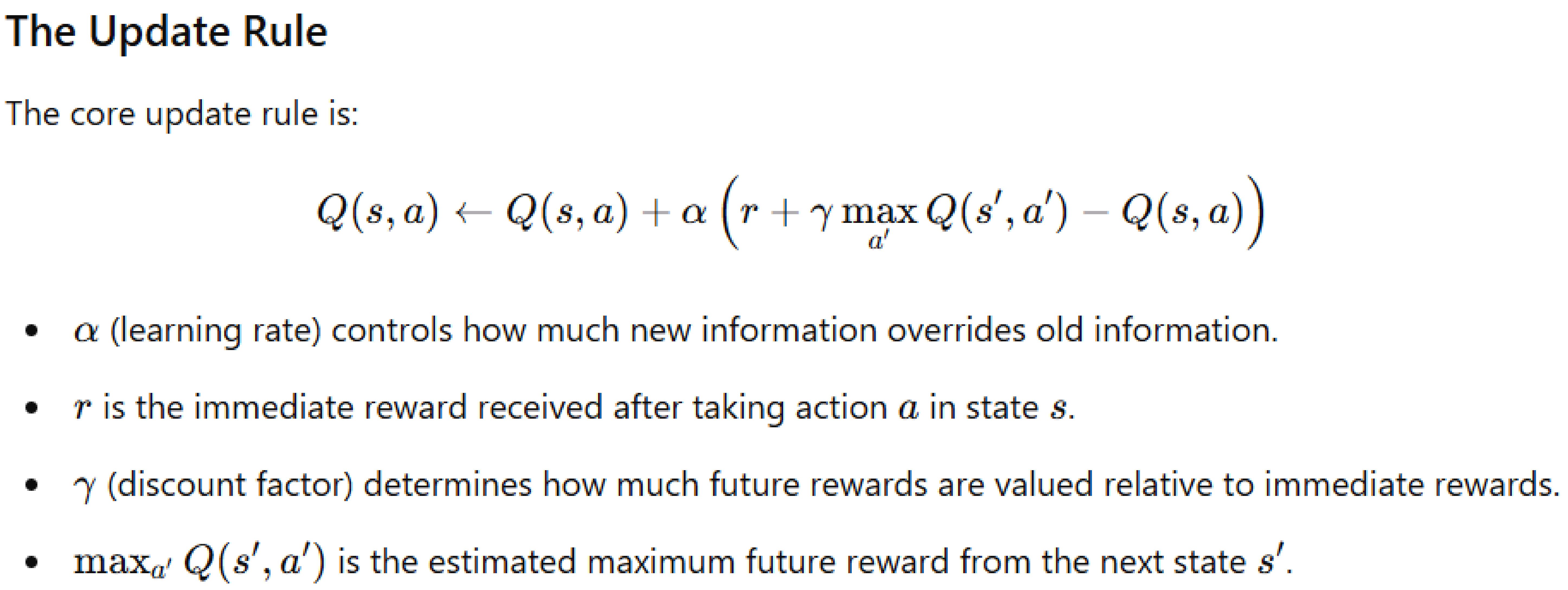
- **Goal:** Reaching the goal gives a reward of +1.
- **Invalid Actions:**
  - Moving out-of-bound or hitting an obstacle gives a reward of -1 and terminates the episode.
- **Neutral Moves:** Valid moves that do not result in hitting an obstacle or reaching the goal provide a reward of 0.

# The Q-Learning Algorithm

## Key Concepts

- **Q-Value:**
  - The Q-value $Q(s, a)$ estimates the expected cumulative reward the agent will receive by taking action $a$ in state $s$ and then following an optimal policy.
- **Q-Table:**
  - A table where each row corresponds to a state and each column corresponds to an action.
  - The goal of Q-learning is to learn this table by iteratively updating it.

# The Q-Learning Algorithm



## The Update Rule

The core update rule is:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left( r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right)$$

- $\alpha$ (learning rate) controls how much new information overrides old information.
- $r$ is the immediate reward received after taking action $a$ in state $s$.
- $\gamma$ (discount factor) determines how much future rewards are valued relative to immediate rewards.
- $\max_{a'} Q(s',a')$ is the estimated maximum future reward from the next state $s'$.

# The Q-Learning Algorithm

## Epsilon-Greedy Exploration

- **Exploration vs. Exploitation:**
  - With probability $\epsilon$, the agent chooses a random action (exploration) to gather more information about the environment.
  - With probability $1 - \epsilon$, the agent chooses the action with the highest Q-value (exploitation) to maximize its reward.
- **Epsilon Decay:**
  - Over time, $\epsilon$ decays, meaning the agent gradually shifts from exploration to exploitation as it becomes more confident in its learned Q-values.