

Lecture 2

Reinforcement Learning

Basic Concepts

jmanero@faculty.ie.edu

MBD-EN2024ELECTIVOS-MBDMCSBT_37E89_467614

- **Key Concepts**
 - **Model**
 - **Policy**
 - **Reward Functions**
- **Delayed and short-term reward**
- **Exploration and Exploitation**
- **Value Learning and Policy Learning**
- **Bellman equation**
- **A Taxonomy of RL approaches**

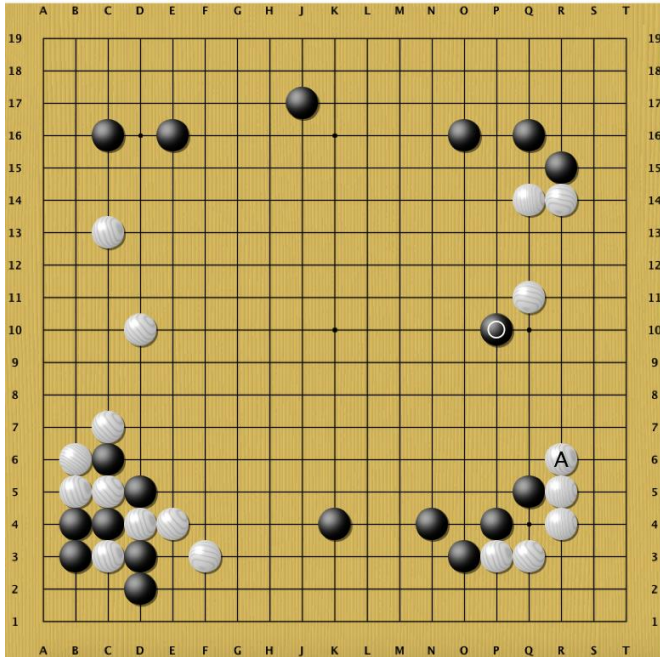
Lee Sedol against AlphaGo



<https://www.wired.com/2016/03/two-moves-alphago-lee-sedol-redefined-future/>

Lee Sedol against AlphaGo

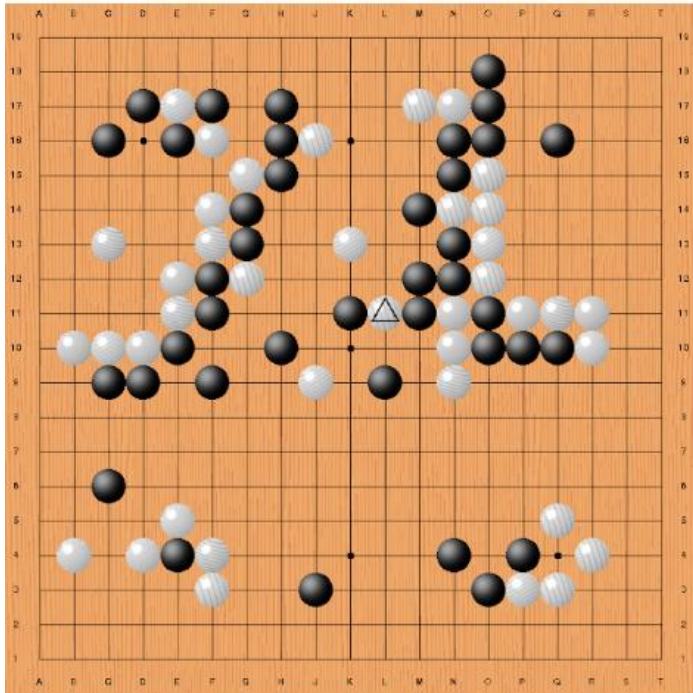
AlphaGo Move 37 in second game was creative and Unique. No human would've ever made



Black 37! This move proved so stunning that, when it appeared on the screen, many players thought the stone had been put down in the wrong place.

Lee Sedol against AlphaGo

Lee Sedol's Move 78 in fourth game was a strange move (it sacrificed stones to create a wedge move)

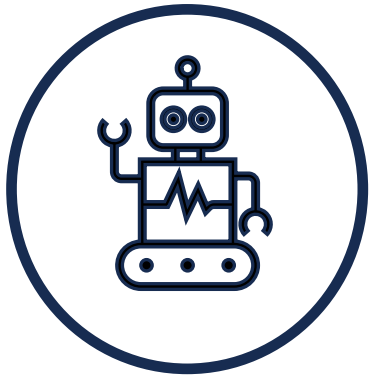


White 78! This move was so unexpected that made AlphaGO to collapse. It was an unlikely movement.

Key Concepts

Key Concepts

Key Concepts: Agent



Agent

AGENT DEFINITION

An Agent is an autonomous computer program

It takes structured actions that can be defined

Key Concepts

Key Concepts: Environment

ENVIRONMENT DEFINITION

Is the world where the Agent exists and operates or interacts



Environment

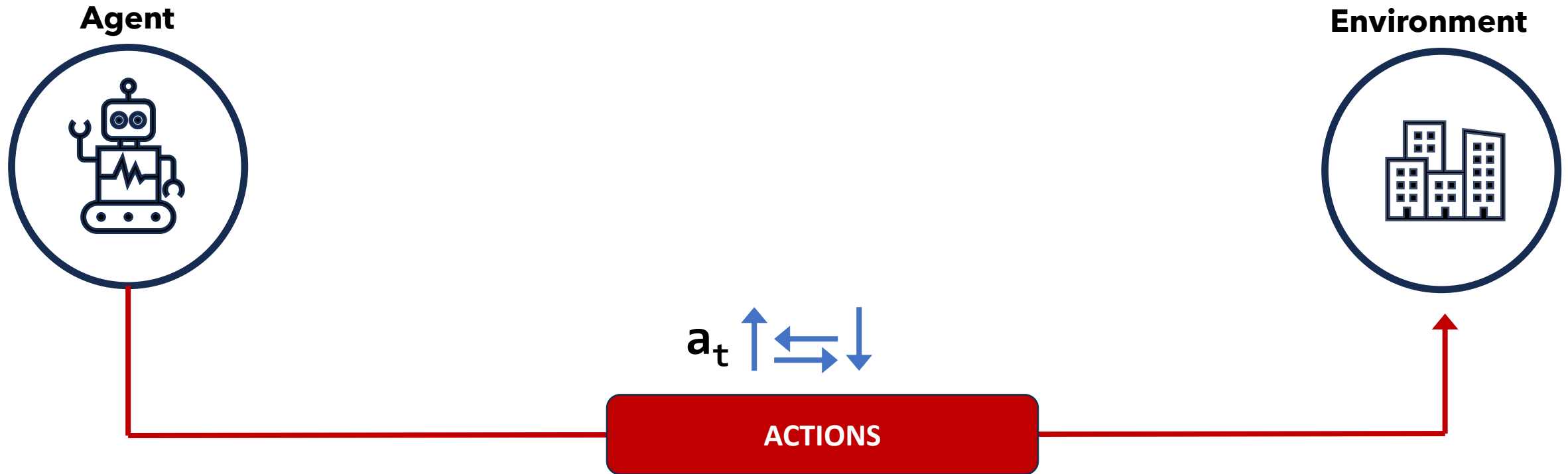
Can be continuous or discrete

Example:

- Chess-discrete
- Robot arm-continuous

Key Concepts

Key Concepts: Actions



ACTION DEFINITION

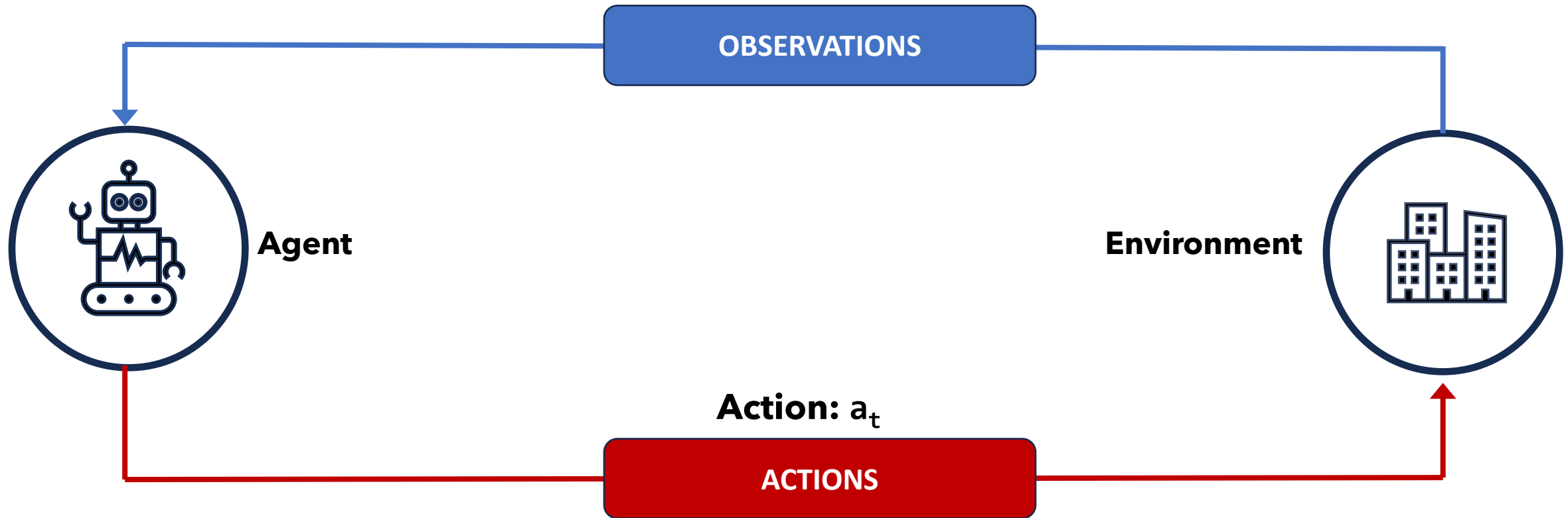
A move the agent can make in the Environment

Action Space **A**: The set of possible actions an agent can perform in the environment

$$\mathbf{A} = \{a_1, a_2, \dots, a_n\}$$

Key Concepts

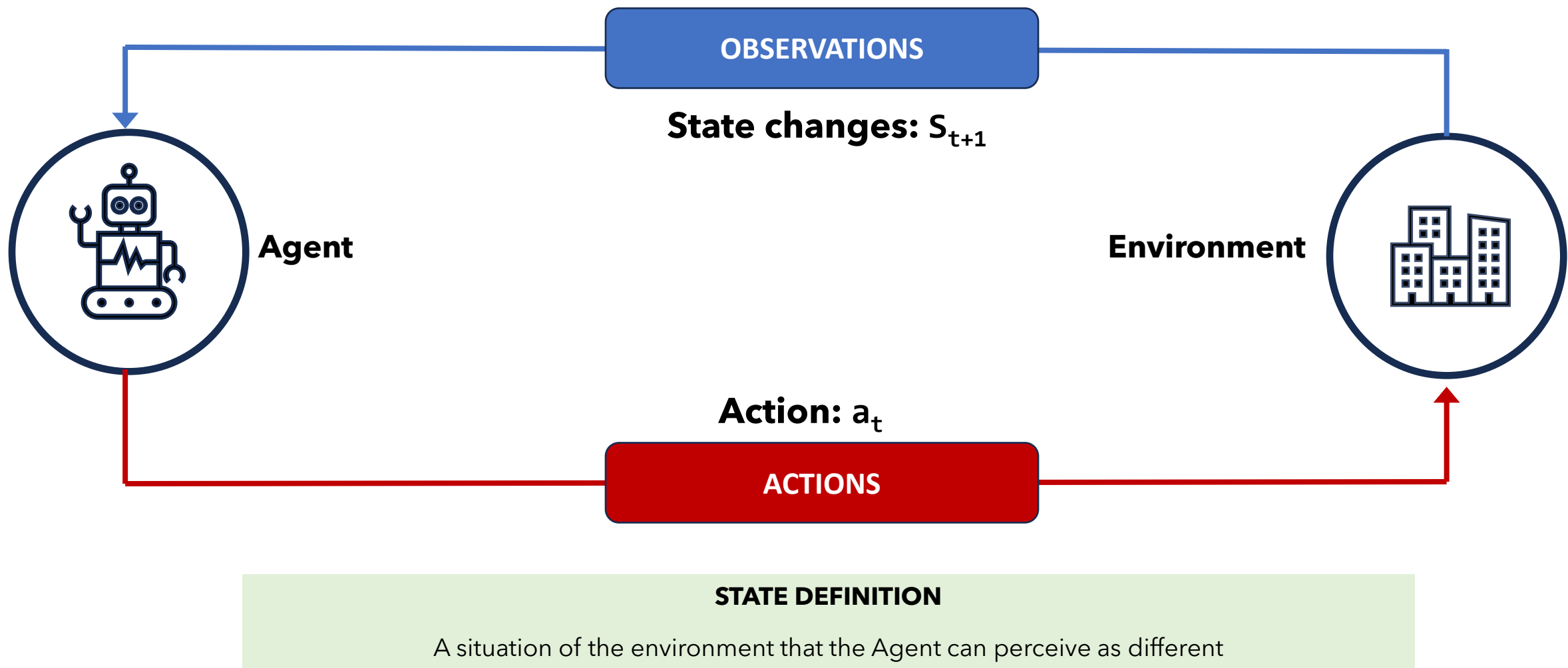
Key Concepts: Observations



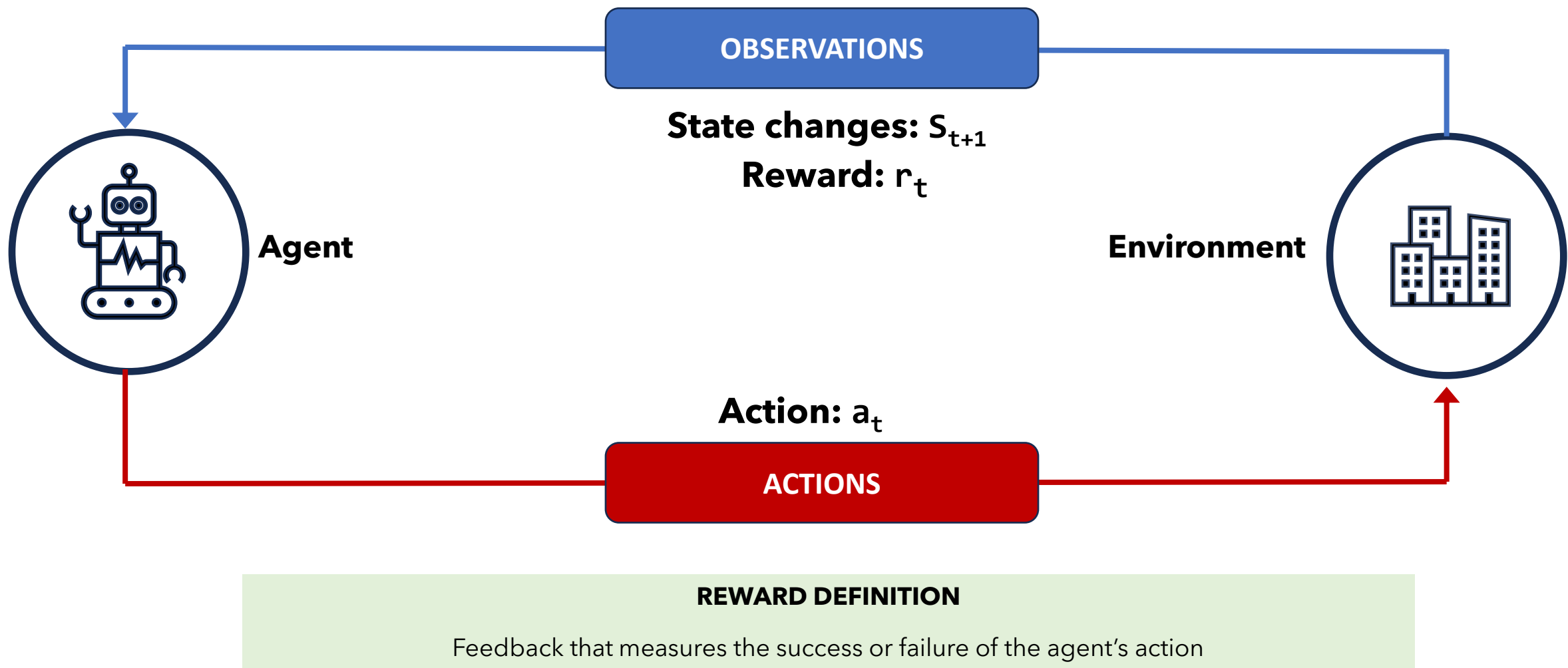
OBSERVATION DEFINITION

Understand the environment after taking actions
(What has changed from last observation, what is new, ...)

Key Concepts
Key Concepts: State and State change

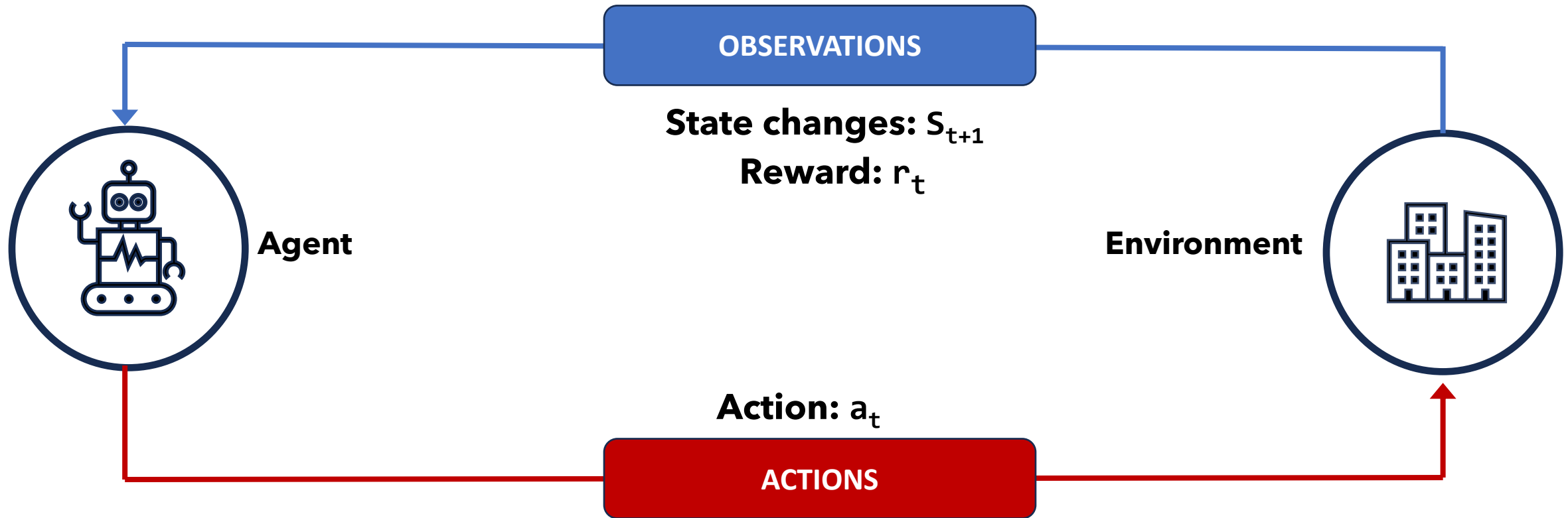


Key Concepts
Key Concepts: Reward



Key Concepts

Key Concepts: Total Reward



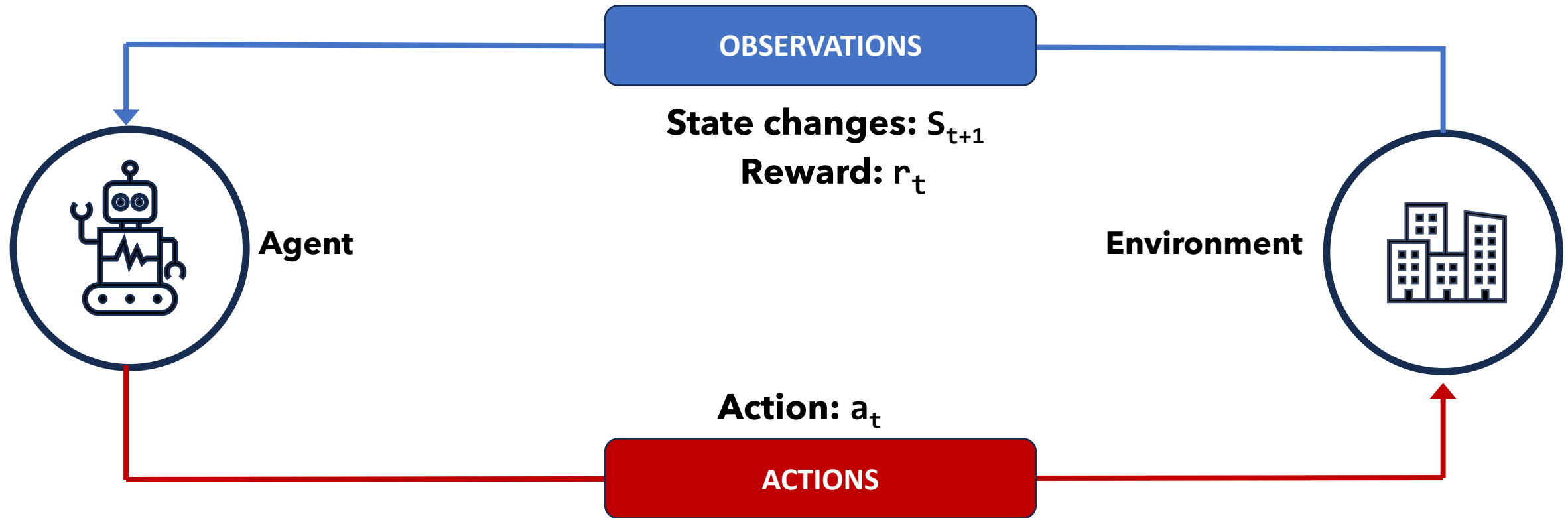
TOTAL REWARD (Return) DEFINITION

Is the Summarization of the total rewards pending until the end of the movement in the universe

$$R_t = \sum_{i=t}^{\infty} r_i$$

Key Concepts

Key Concepts: Total Reward decomposition



DEFINITION

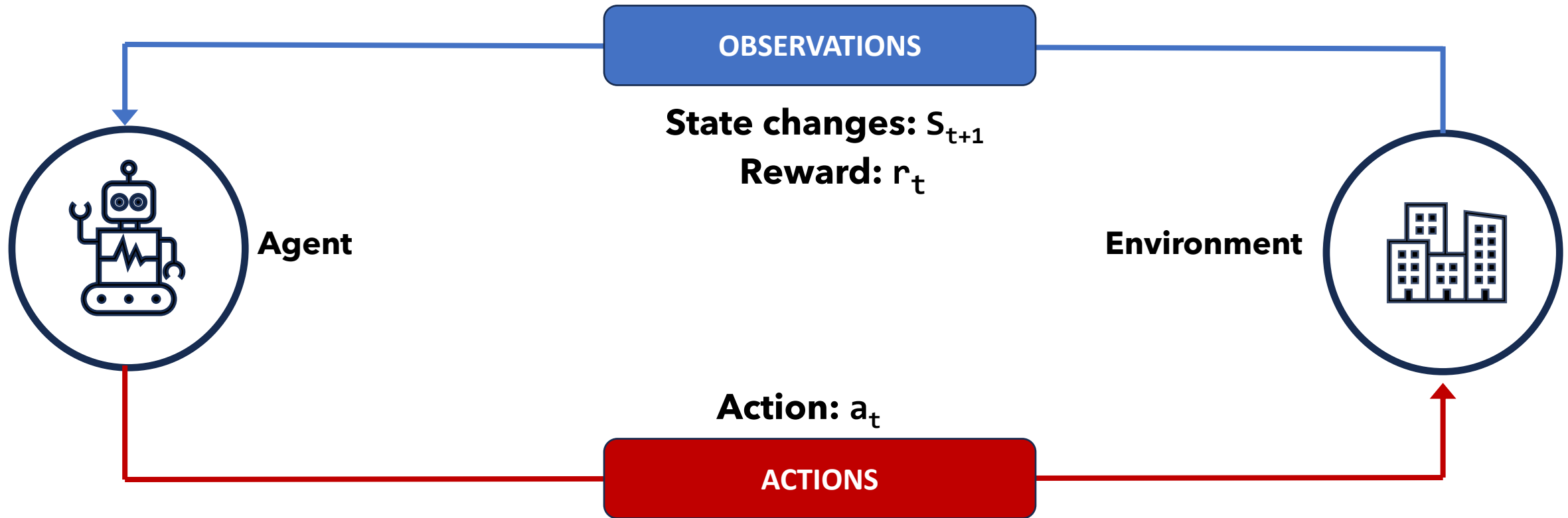
TOTAL REWARD (Return)

Is the Summarization of the total rewards pending until the end of the movement in the universe

$$R_t = \sum_{i=t}^{\infty} r_i = r_t + r_{t+1} + \dots + r_{t+n} + \dots$$

Key Concepts

Key Concepts: Discounted Total Reward



DEFINITION

Discount is a value between 0 and 1

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i$$

Delayed and short-term rewards

Key Concepts

Hold on: What is the difference between Total Reward and Discounted Total Reward?

The problem with long term rewards. A Parable of a Chinese farmer



Horse and Groom, after Li Gonglin 臨李公麟〈人馬圖〉 Smithsonian National Museum of Asian Art (Washington USA)

Hold on: What is the difference between Total Reward and Discounted Total Reward?

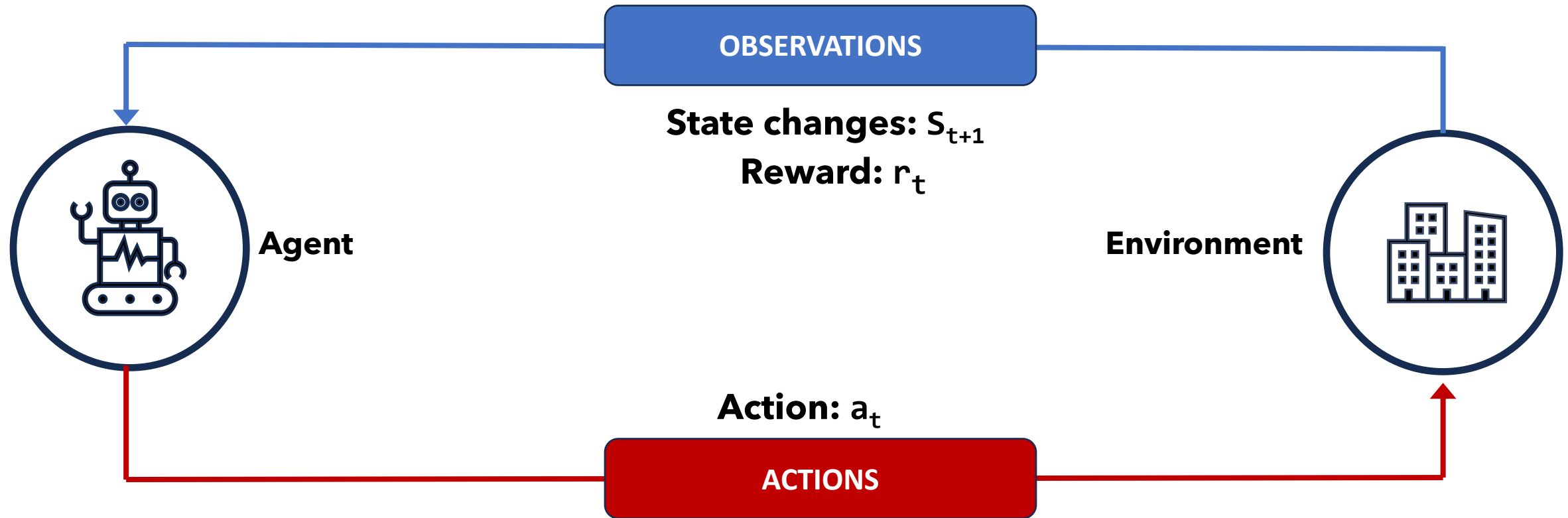
The reward value is time dependent.

- For instance. if we are playing chess the intermediate rewards are only important if they help us to approach the Final goal which is Wining the game.
- How to weight the short-term rewards and the long-term rewards is a complex decision.
 - If we put too much reward in the short-term goal, the agent may be misled in his path to the goal.
 - If we weight too much the long-term rewards, maybe the actual rewards will end up not having any relationship with the final goal/reward.

The main difference between discounted total reward and total reward lies in how future rewards are valued: discounted total reward places greater emphasis on immediate rewards and diminishes the value of future rewards, while total reward treats all rewards equally, regardless of when they occur. This distinction has significant implications for the agent's learning and decision-making process

Key Concepts

Key Concepts: Total Reward decomposing discount factor



DEFINITION

Discount factor gamma is a value between 0 and 1

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \dots + \gamma^{t+n} r_{t+n} + \dots$$

$$0 < \gamma < 1$$

Long Term Rewards

Delayed consequences issues

Decisions now can impact things much later...

- Saving for retirement
- Finding a key in video game Montezuma's revenge

Introduces two challenges

- When planning: decisions involve reasoning about not just immediate benefit of a decision but also its longer-term ramifications
- When learning: temporal credit assignment is hard (what caused later high or low rewards?)

The problem with delayed and long-term rewards

- Short term rewards are easy to calculate and learn
- What happens when the reward only happens at the end of a long sequence of actions?
- If the number of states is very large long-term impact of our actual actions can be quite difficult to calculate
- How do we assign as immediate reward from a long-term reward using discount factor impact the learning of the agent
- It may be better to sacrifice immediate reward to gain more long-term reward
- Two challenges
 - Decisions on next action involve reasoning not just immediate Benefit of a decision, but also in longer term ramifications
 - Assigning temporal credit when learning is hard (to link the actual credit with future actions)

Exploration and Exploitation

Exploration and exploitation

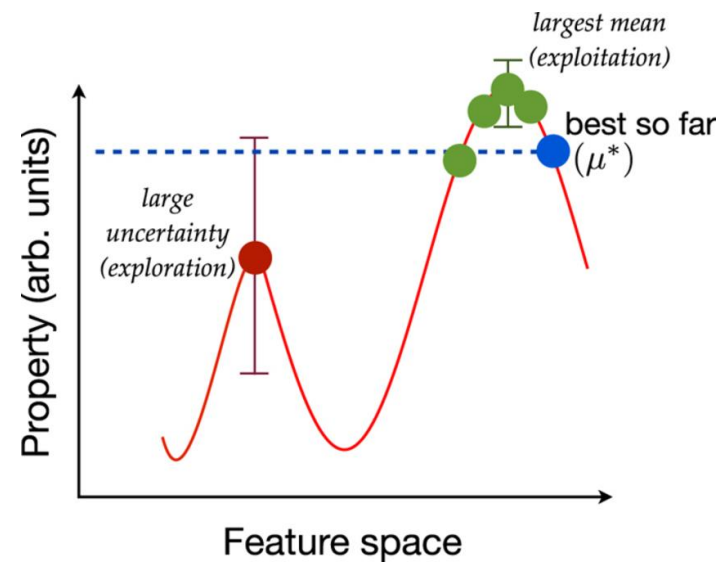
Two important concepts

The **exploration-exploitation dilemma**, also known as the **explore-exploit tradeoff**, is a fundamental concept in decision-making that arises in many domains.

Consists of a balancing act between two opposing strategies. Exploitation involves choosing the best option based on current knowledge of the system (which may be incomplete or misleading), while exploration involves trying out new options that may lead to better outcomes in the future at the expense of an exploitation opportunity.

How to balance it?

Exploration and Exploitation

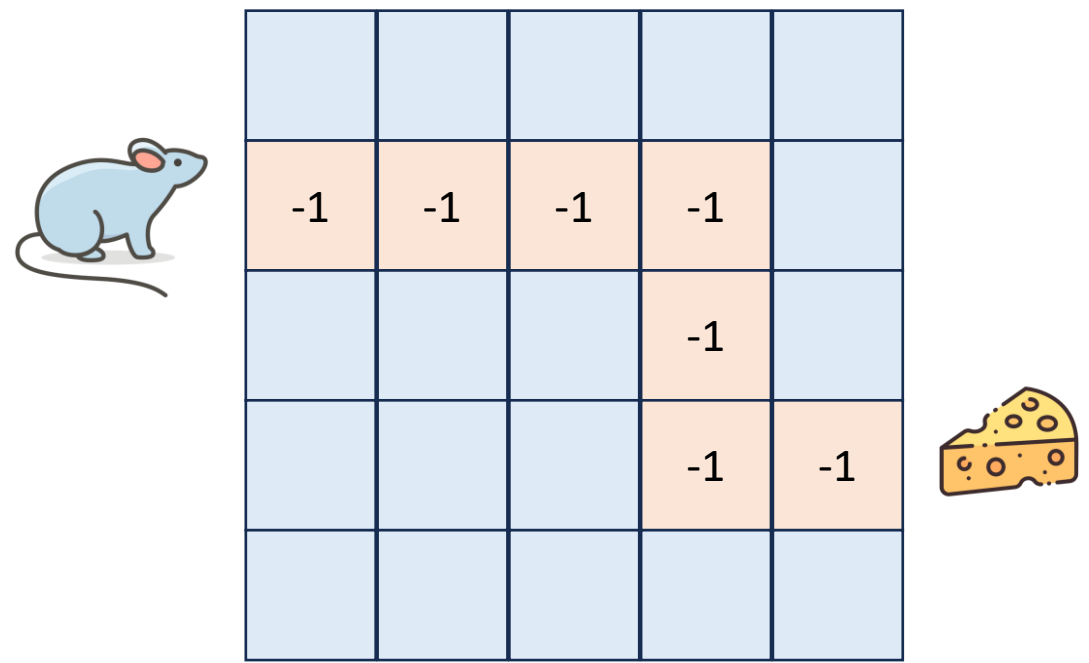


HIGH	Uncertainty	LOW
Search and breakthrough	Focus	Efficiency and growth
Many small bets, expecting few outsized winners	Financial philosophy	Safe haven with steady returns and dividends
Iterative experimentation, embracing speed, failure, learning and rapid adaption	Culture and process	Linear execution, embracing planning, predictability, and minimal failure
Explorers who excel in uncertainty	People and skills	Managers who are strong at organizing and planning

Artificial Environment

Artificial Environment

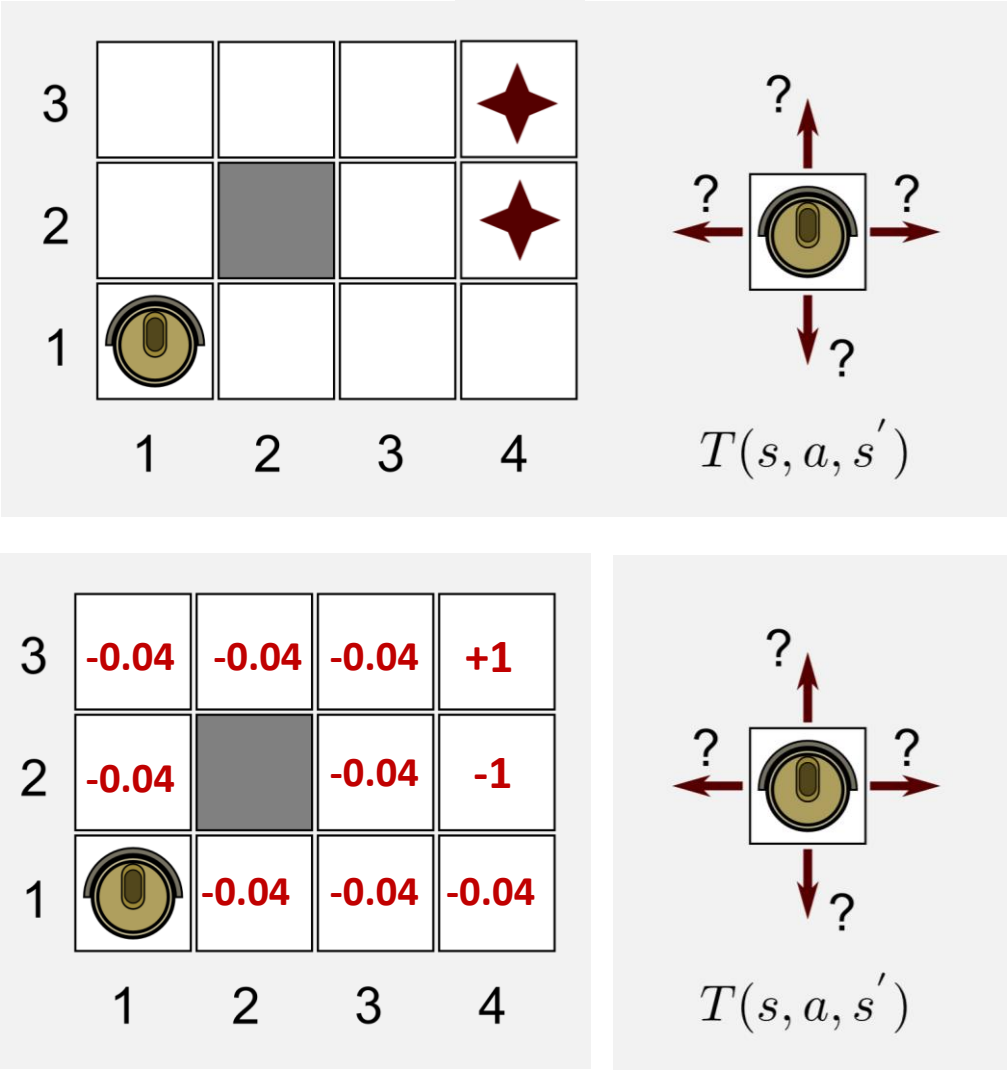
What is an artificial Environment?



Action: UP, DOWN, LEFT, RIGHT
Reward: Each move = -1

Artificial Environment

Russell's World



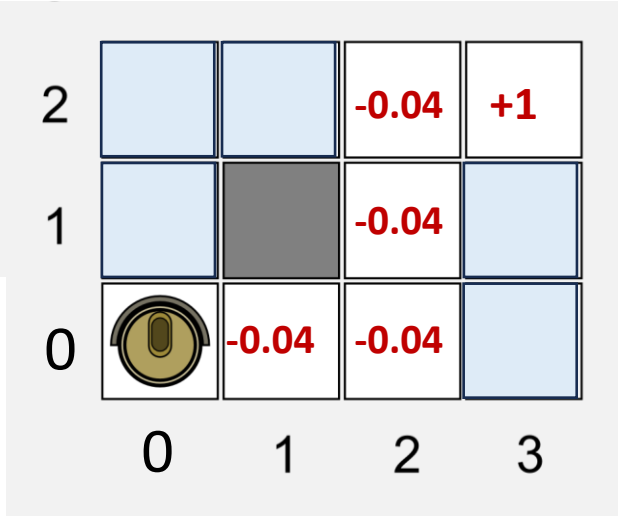
Environment and movements

- Set of possible States: $S=\{s_0,s_1,..., s_m\}$
- Initial State: s_0
- Set of possible Actions: $A=\{a_0,a_1,...,a_n\}$

Rewards

Every move **-0.04**
Goal **+1**
Bad Goal **-1**

Artificial Environment
Example total Reward



Reward: ????

Policy Learning and Value Learning

Policy Learning & Value Learning

Defining the Q function / Total Reward

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i = \gamma^t r_t + \gamma^{t+1} r_{t+1} + \dots + \gamma^{t+n} r_{t+n} + \dots$$

Total reward, R_t is the discounted sum of all rewards obtained from time t

$$Q(s_t, a_t) = E[R_t \mid s_t, a_t]$$

The Q-function tells us the Agent **expected total future reward** an agent has in state \mathbf{s} in the instant \mathbf{t} if it executes action \mathbf{a}

Policy Learning & Value Learning

Q function – Think of it as a table

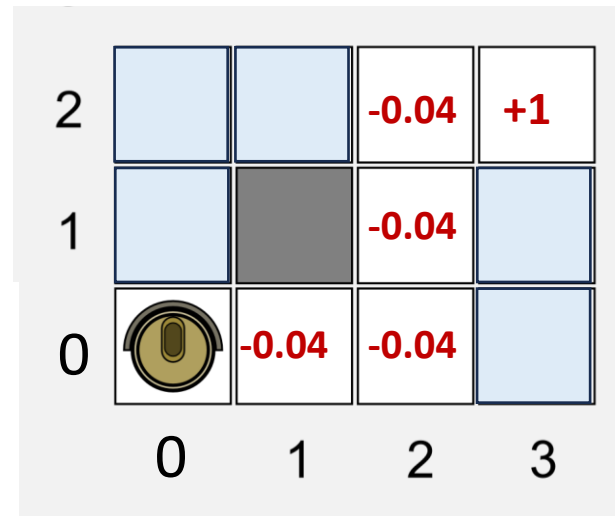
$$\gamma = 0.9$$

$$t = 0$$

$$n = 5$$

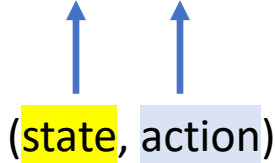
$$Q(0,0) = \gamma^0 r_t + \gamma^1 r_{t+1} + \dots + \gamma^5 r_{t+n}$$

$$Q(0,0) = 0 - 0.99 \cdot 0.04 - 0.99^2 \cdot 0.04 - 0.99^3 \cdot 0.04 - 0.99^4 \cdot 0.04 + 0.99^5 \cdot 0.04 = -0.118$$



Policy Learning & Value Learning

The concept of Policy

$$Q(s_t, a_t) = E[R_t \mid s_t, a_t]$$


(state, action)

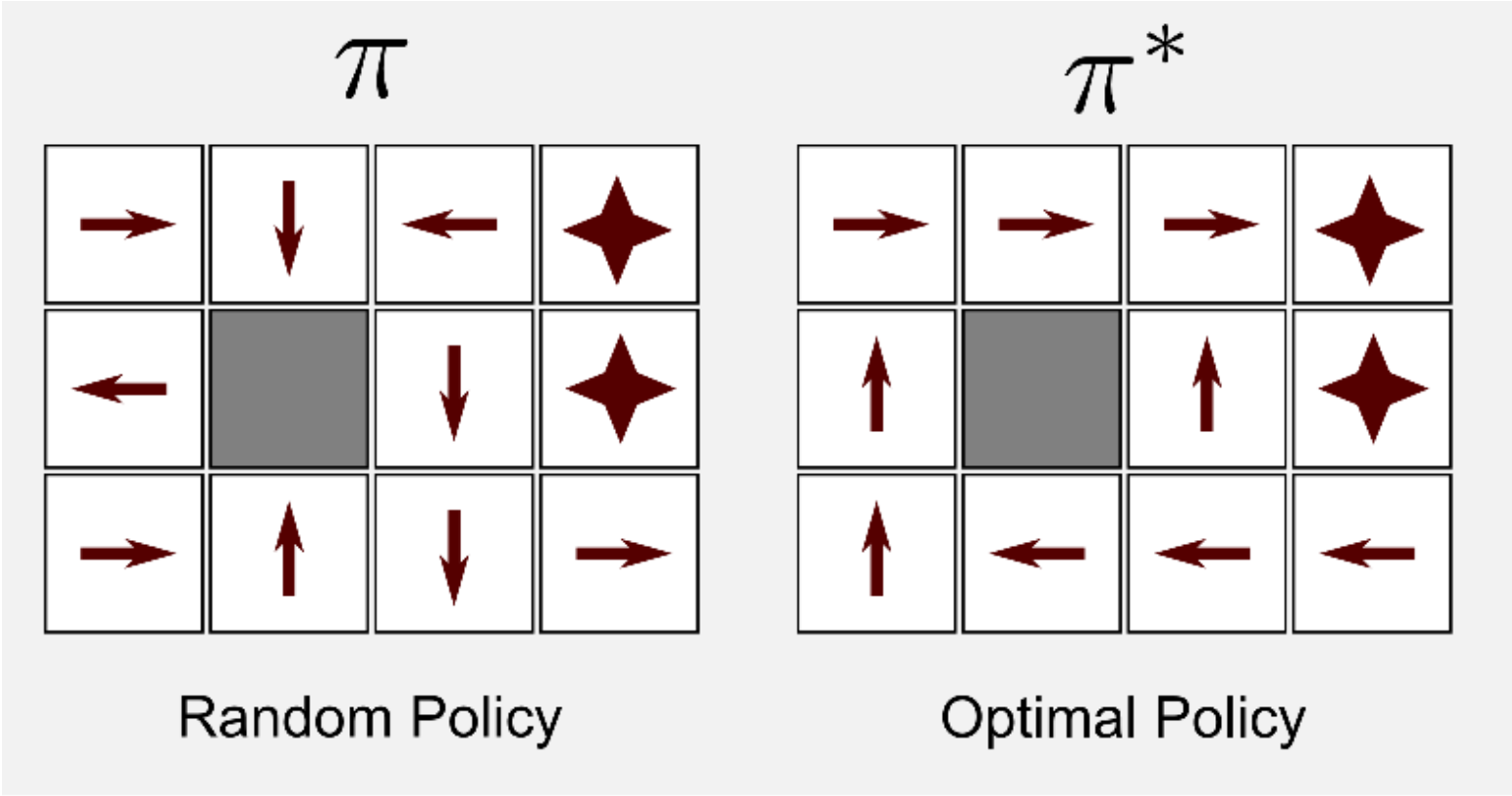
The agent needs a policy $\pi(s)$, to infer the best action to take at its state, s

The policy should choose an action that maximizes future reward

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} Q(s, a)$$

Policy Learning & Value Learning

The concept of Policy



Reinforcement Learning

Two groups of RL Algorithms

Policy Learning

Find $\pi(s)$

Sample $a \sim \pi(s)$

Policy Learning tries to optimize the policy function to maximize rewards

Value Learning

Find $Q(s, a)$

$a = \underset{a}{\operatorname{argmax}} Q(s, a)$

Value Learning obtains the value function for all states and applies it to navigate the universe

Bellman Equation

Bellman Equation

Markov Process (no rewards)

- A Markov Process is a memoryless random Process
 - A sequence of random states ($s \in S$)
- Definition
 - **S** is a (finite) set of states
 - **P** is a dynamic/transition model that specifies $p(s_{t+1} = s' | s_t = s)$
- With a finite number of states, we can express **P** as a matrix

$$P = \begin{pmatrix} P(s_1|s_1) & P(s_2|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & P(s_2|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_1|s_N) & P(s_2|s_N) & \cdots & P(s_N|s_N) \end{pmatrix}$$

Bellman Equation

Markov Reward process

- A Markov Reward Process is a Markov Chain + Rewards
 - **S** is a finite set of states ($s \in S$)
 - **P** is a dynamic/transition model that specifies $p(s_{t+1} = s' | s_t = s)$
 - **R** is a reward function $R(s_t = s) = \mathbb{E}[r_t | s_t = s]$
 - Discount factor $\gamma \in [0, 1]$
- Horizon
 - Number of time steps in each episode
 - Could be infinite
 - Return of a MRP is

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{H-1} r_{t+H-1}$$

Bellman Equation

Value Function and Bellman equation

State Value Function $V(s)$ from a MRP is

$$V(s) = \mathbb{E}[G_t | s_t = s] = \mathbb{E}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots + \gamma^{H-1} r_{t+H-1} | s_t = s]$$

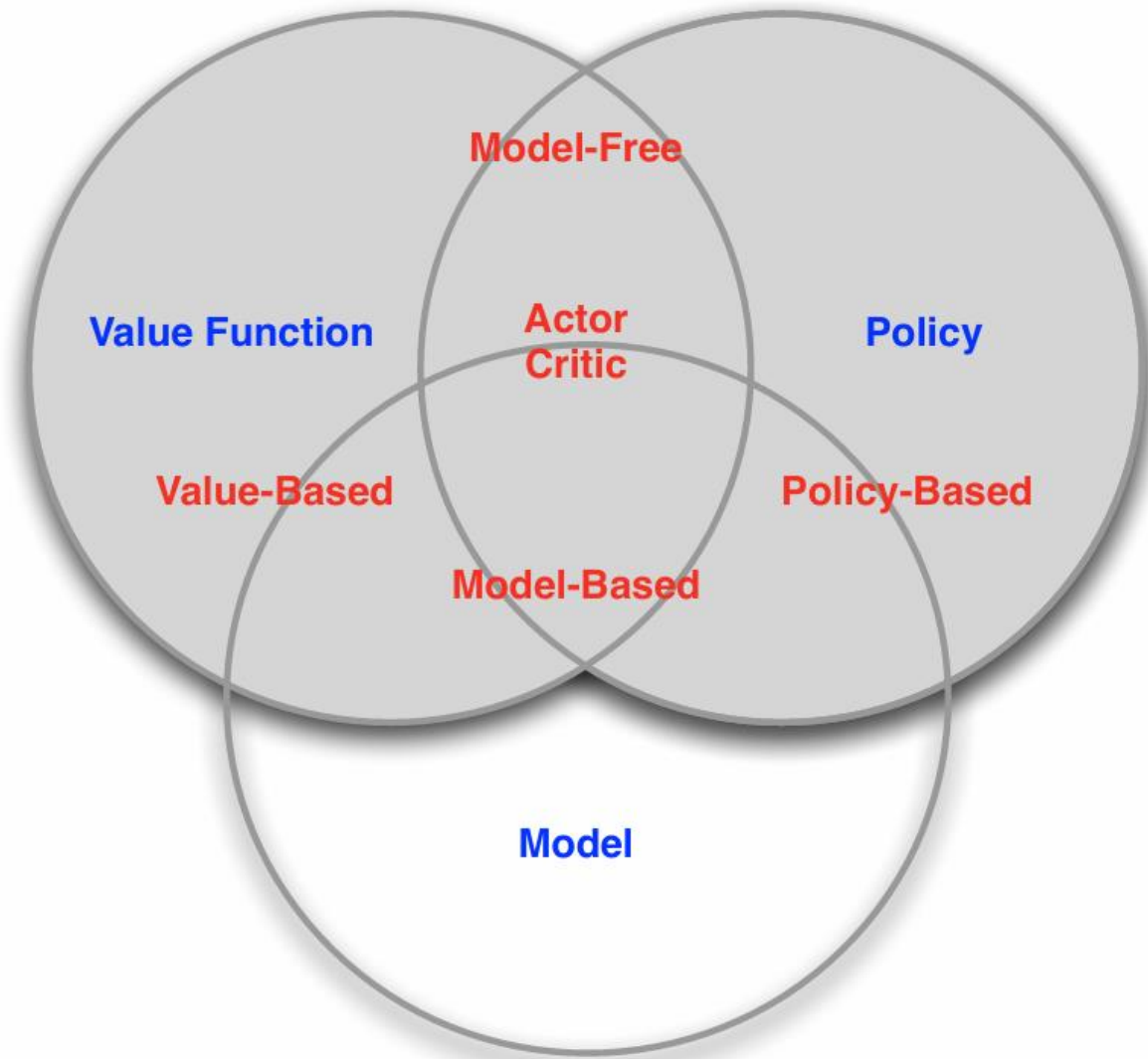
Bellman Equation

$$V(s) = \underbrace{R(s)}_{\text{Immediate reward}} + \underbrace{\gamma \sum_{s' \in \mathcal{S}} P(s'|s) V(s')}_{\text{Discounted sum of future rewards}}$$

A Taxonomy of Approaches

Taxonomy of RL Agents

Combination of Value Function/ Policy and Model

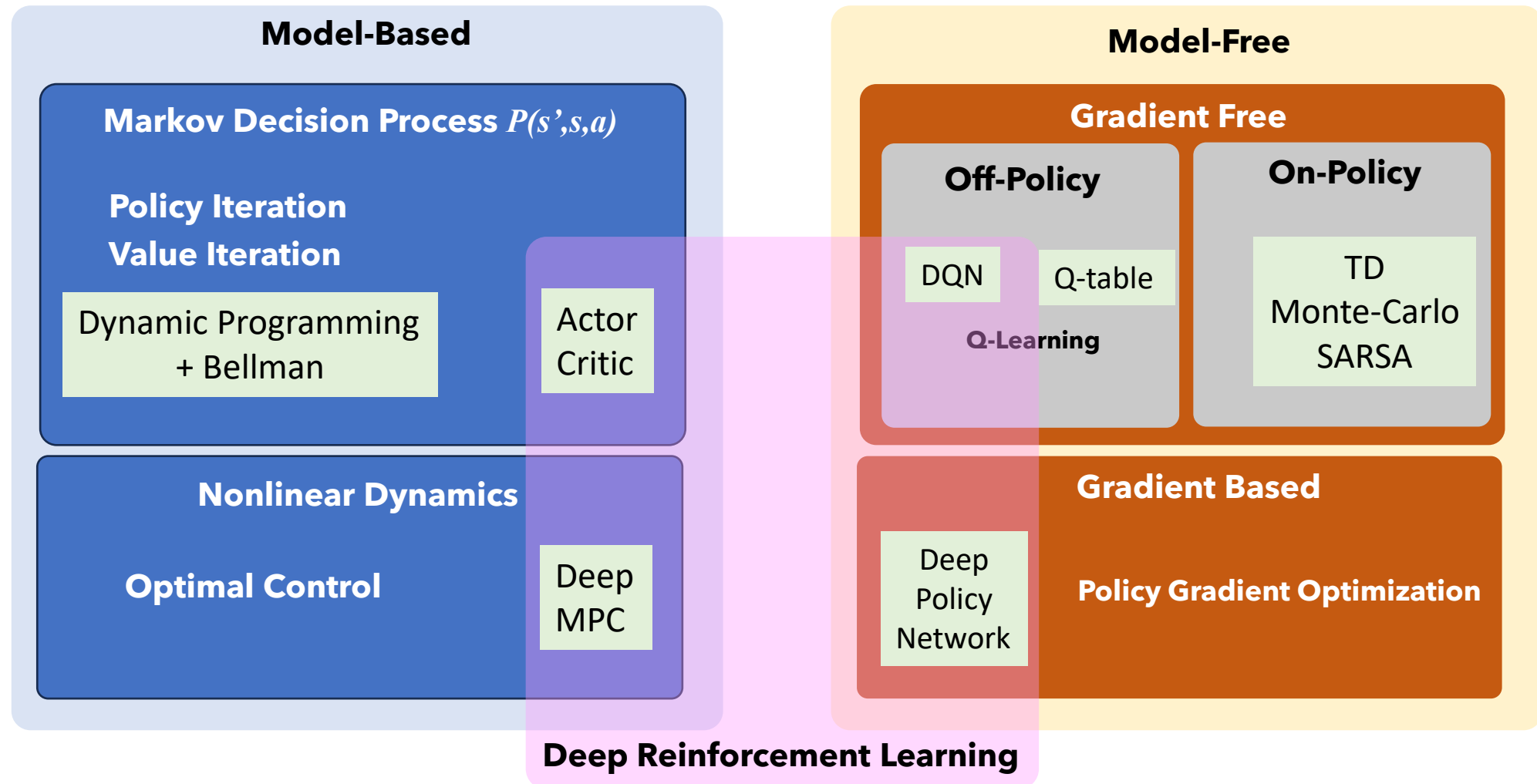


Taxonomy of RL Agents

(*) From David Silver RL Course UCL

A Taxonomy of RL Methods

Classical classification by David Silver



A Taxonomy of RL Methods

What is Model-based and Model-Free?

Model-based

- The agent tries to create a Map of the environment
- They use a policy function to guide the choices
- The policy helps to take the decision for the Best Next Action

- Learn Faster
- When the problem has clear outcomes and results is better
- Less risky in critical systems

Model-Free

- The agent learns by trial and error
- There is no policy function
- No knowledge of the environment

- Better when there are lots of data
- Computing intensive
- Better with uncertain situations

A Taxonomy of RL Methods

What is Model-based and Model-Free?

- **Suitability for Different Scenarios** Model-based and model-free reinforcement learning each shine in different situations. They fit various scenarios based on their unique strengths and weaknesses.
- **Fast Learning Needs:** Model-based approaches often learn faster than model-free methods. This makes them better for tasks where quick learning from limited data is crucial, like dexterous manipulation.
- **Data-Rich Environments:** Model-free methods are great when there's lots of data to learn from. These algorithms, like deep Q-learning, can handle complex environments with many details.
- **Computation Resources:** If you've got powerful computers at your disposal, model-free learning can take full advantage. It uses more computing power to analyze loads of data without needing a predefined model.
- **Predictable Outcomes:** When tasks have clear results that don't change much, model-based learning works well. It predicts future events based on past experiences.
- **Uncertain Situations:** Model-free is good when things are unpredictable. Since it doesn't assume anything about the future, it can adjust easily to new information or changes in the environment.
- **Real-world Tasks:** For real jobs like controlling robots or self-driving cars, model-based systems can be less risky. They plan ahead using a model of the world which can help avoid mistakes.

What is Gradient Free and Gradient Methods?

Gradient

- Use a gradient optimization to maximize the objective function
- Are efficient
- Not all problems have smooth gradients (noise-discontinuities)
- Problem of local minima

Gradient-Free

- Work in noisy environments
- They may find the global minimum without getting trapped in local minima
- Inefficient for large parameter spaces
- They don't understand how parameter change impacts optimization

Wrap-up

- **Reinforcement Learning Key Concepts:** To model RL algorithms we define the key components
 - **Model:** Model is defined by a set of actions and observational space
 - **Policy:** Is the strategy that the agent will follow in the environment to accomplish a task
 - **Reward:** incentives (positives or negative) to move in the environment
- Delayed rewards are tricky to calculate and use (even for humans)
- Agents need a combination of exploration and exploitation
- There are two major ways to learn, based on Value or based on Policy
- The sequence of states must be a Markov process
- We define the Bellman equation. Our objective will be to optimize it to maximize value
- There are two major RL groups of algorithms, Model free and Model

END

Lecture 1

