



APRENDIZADO POR REFORÇO

Aula 1: Introdução

Lucas Pereira Cotrim

Marcos Menon José

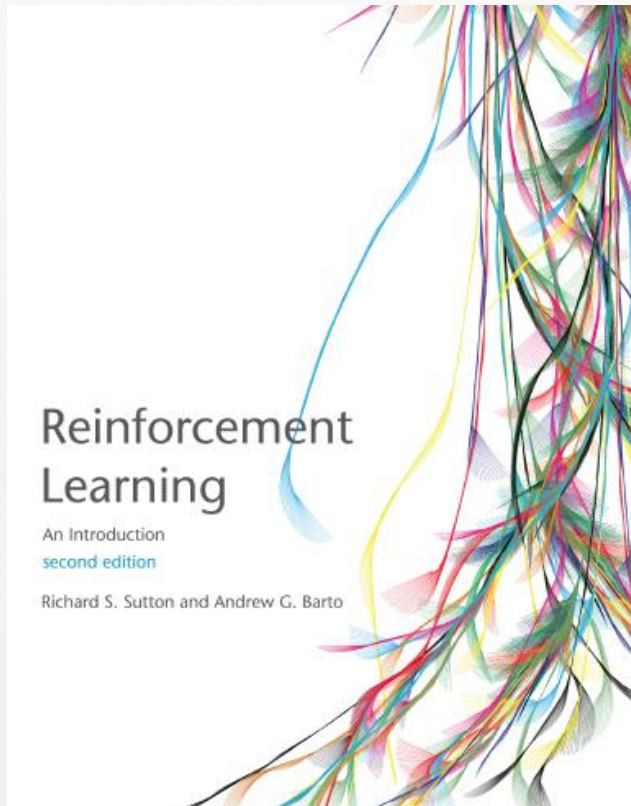
lucas.cotrim@maua.br

marcos.jose@maua.br

APRESENTAÇÃO DO CURSO

- **Livro Texto:**

Sutton, R. S. and Barto, A. G. *Reinforcement Learning: An Introduction*, A Bradford Book, Cambridge, MA, The MIT Press, 2018. (Edição Original 1998)



<http://incompleteideas.net/book/RLbook2020.pdf>

APRESENTAÇÃO DO CURSO

- **Outra Referência:**

Szepesvári , C. *Algorithms for Reinforcement Learning*. Morgan & Claypool Publishers, 2010.

Algorithms for Reinforcement Learning

Draft of the lecture published in the
Synthesis Lectures on Artificial Intelligence and Machine Learning
series
by
Morgan & Claypool Publishers

Csaba Szepesvári

June 9, 2009*

Contents

1 Overview	3
2 Markov decision processes	7
2.1 Preliminaries	7
2.2 Markov Decision Processes	8
2.3 Value functions	12
2.4 Dynamic programming algorithms for solving MDPs	16
3 Value prediction problems	17
3.1 Temporal difference learning in finite state spaces	18
3.1.1 Tabular TD(0)	18
3.1.2 Every-visit Monte-Carlo	21
3.1.3 TD(λ): Unifying Monte-Carlo and TD(0)	23
3.2 Algorithms for large state spaces	25
3.2.1 TD(λ) with function approximation	29
3.2.2 Gradient temporal difference learning	33
3.2.3 Least-squares methods	36

*Last update: March 12, 2019

APRESENTAÇÃO DO CURSO

- **Avaliação:**

$$M = \frac{7\bar{T} + 3\bar{E}_x}{10} + 0.05 E_{extra}$$

- $\bar{E}_x = \frac{E_1 + E_2 + E_3}{3}$: Média dos Exercícios.
- $\bar{T} = \frac{T_1 + T_2 + T_3 + T_4}{4}$: Média dos Trabalhos.
- E_{extra} : Exercício Extra.




Página inicial / Meus Cursos / Pós-Graduação / TIA-2021/1



Meus Cursos



TIA-2021/1-Técnicas Avançadas de IA e Computação


 Alterar a imagem da capa

CONTEÚDO



Introdução

Aprendizado por Reforço

 Crie uma nova seção

 Painel do Curso

Aprendizado por Reforço



Use esta área para descrever sobre o tópico - com texto, imagens, áudio e vídeo.



Editar seção

PASTA

Aula 1 - Introdução



PASTA

Aula 2 - MDP



PASTA

Aula 3 - DP, Model-Free Prediction





APRESENTAÇÃO DO CURSO

- Cronograma:**

Aula	Data	Prof.	Tópico	Entrega
1	15/03	Lucas e Marcos	Introdução, E_{extra}	-
2	22/03	Lucas	Markov Decision Process (MDP), E_1	-
3	29/03	Lucas	Model-Free Prediction (Tabular Methods), E_2	E_{extra}
4	05/04	Marcos	Model-Free Control (Tabular Methods), E_3	E_1
5	12/04	Marcos	Function Approximation and DRL, T_1	E_2
6	19/04	Lucas	Policy Gradient Methods, T_2	E_3
7	26/04	Lucas	Model-Based Methods, T_3	T_1
8	05/03	Marcos	Métodos Modernos e Estudo de Caso, T_4	T_2
				T_3
				T_4

APRENDIZADO POR REFORÇO: INTUIÇÃO

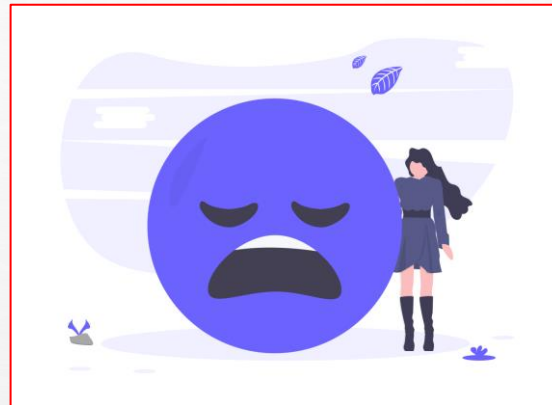
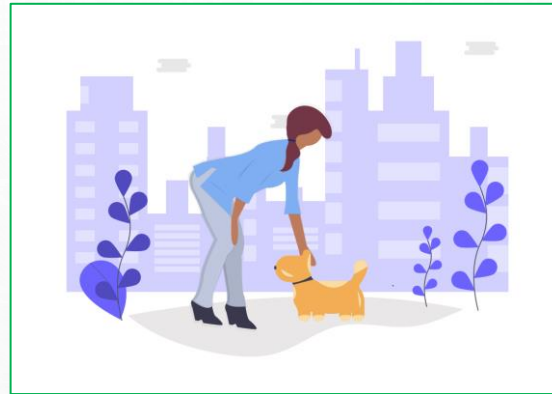
estado

ação

recompensa

brincar

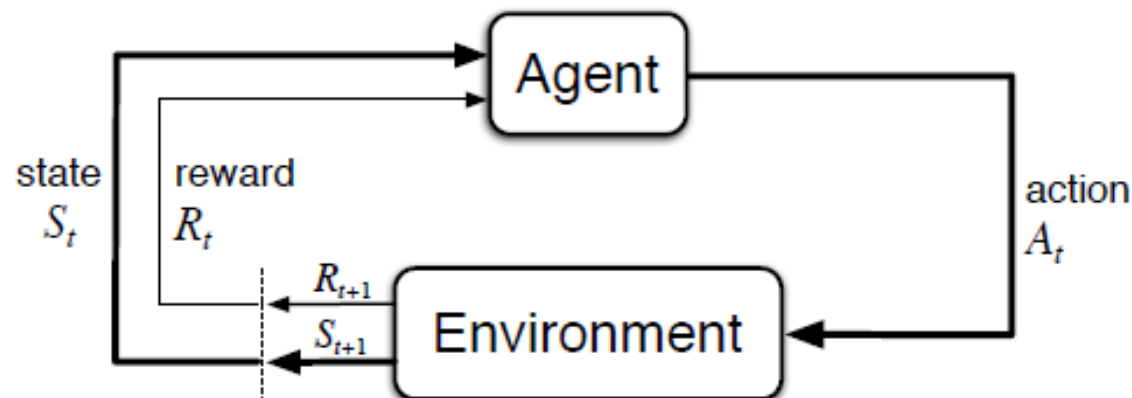
morder



- **Interação com ambiente:**
Diferentes ações do agente levam a diferentes resultados.
- **Recompensas:** Agente recebe recompensa (ou penalidade) em função da ação tomada.
- **Objetivo do Aprendizado por Reforço:** Maximizar recompensas obtidas.

APRENDIZADO POR REFORÇO: DEFINIÇÃO

- Área de Aprendizado de Máquina associada a como agentes devem escolher ações em determinado ambiente com o objetivo de maximizar recompensas.
- Características do problema:
 - Agente possui sensores que observam o **estado** do **ambiente**.
 - O agente possui um conjunto de possíveis **ações** que pode tomar para alterar esse estado.
 - A cada ação tomada ele recebem uma **recompensa** que indica a qualidade da ação dado o estado.

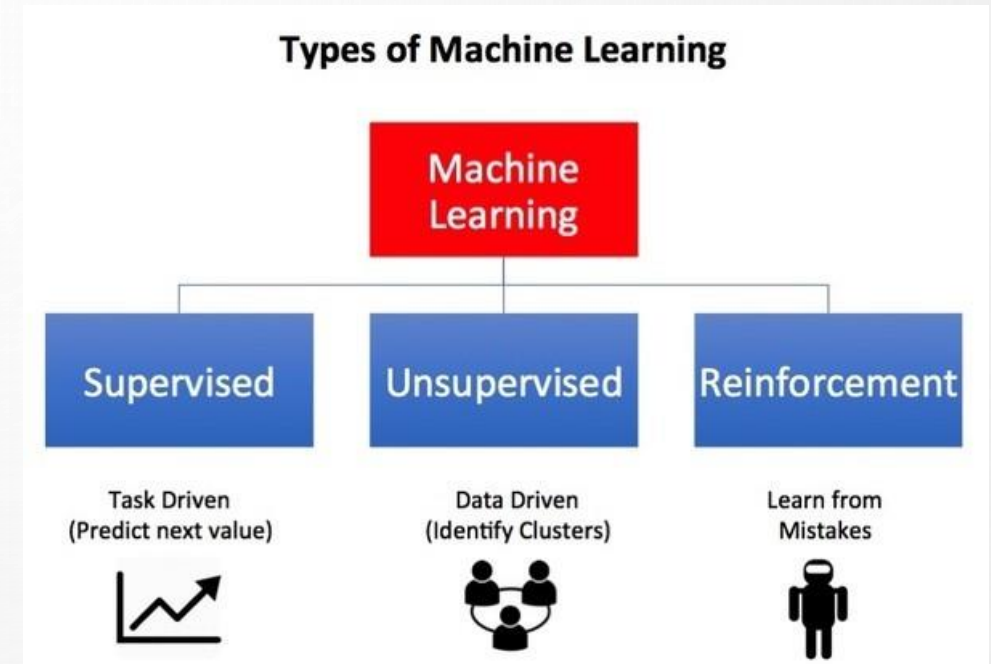


APRENDIZADO POR REFORÇO: COMPARAÇÃO COM OUTRAS ÁREAS

O Aprendizado por Reforço pode ser visto como a terceira grande área de Aprendizado de Máquina.

As três áreas podem ser divididas em função do tipo de *feedback*:

- **Aprendizado Supervisionado:** Respostas corretas (*labels*) para cada amostra.
- **Aprendizado não Supervisionado:** Não apresenta rotulação, efetua agrupamento.
- **Aprendizado por Reforço:** Recompensas esporádicas.



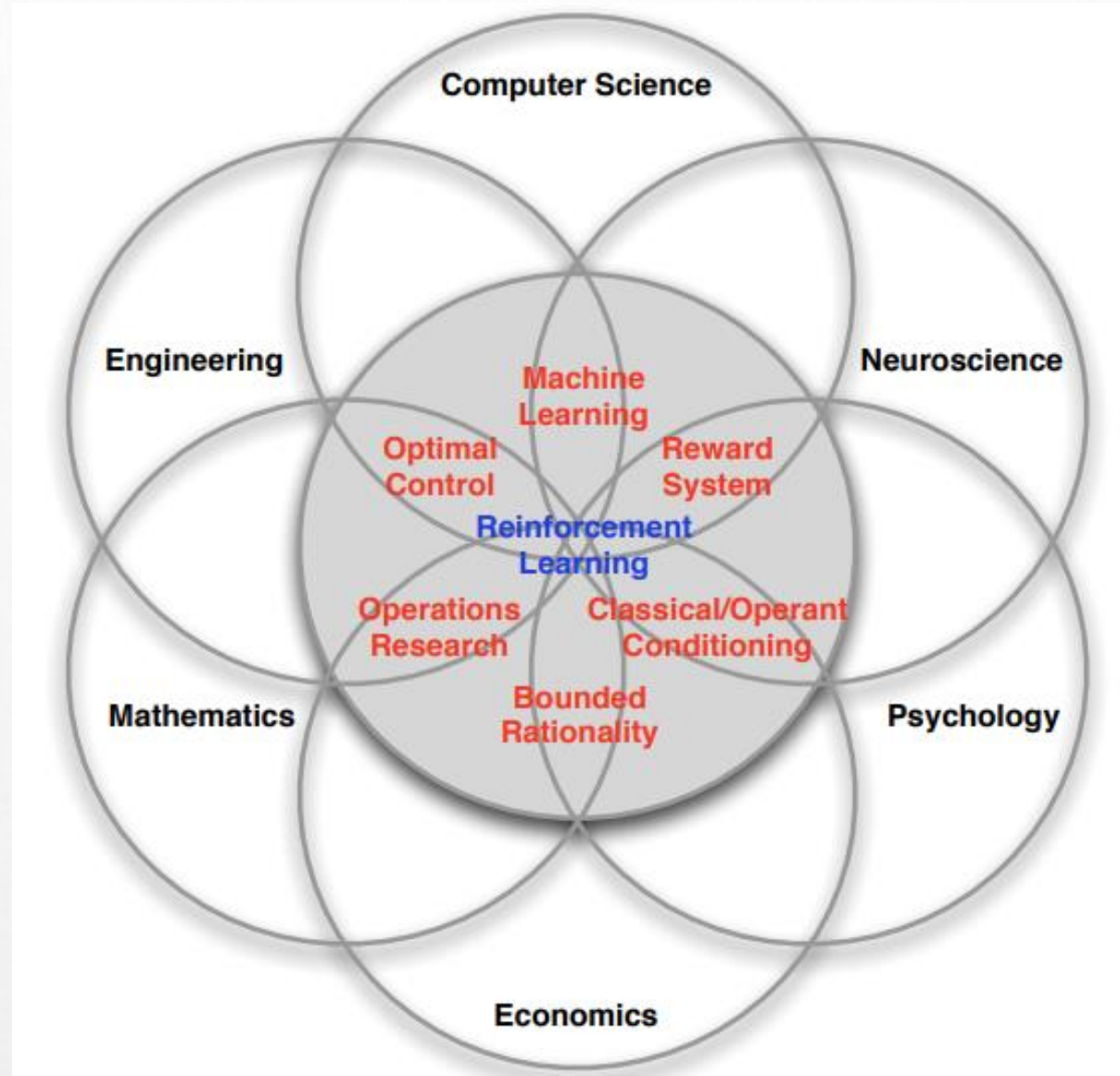
Fonte: <https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f>

APRENDIZADO POR REFORÇO: COMPARAÇÃO COM OUTRAS ÁREAS

O Aprendizado por Reforço apresenta características diferentes do paradigma tradicional de Aprendizado de Máquina:

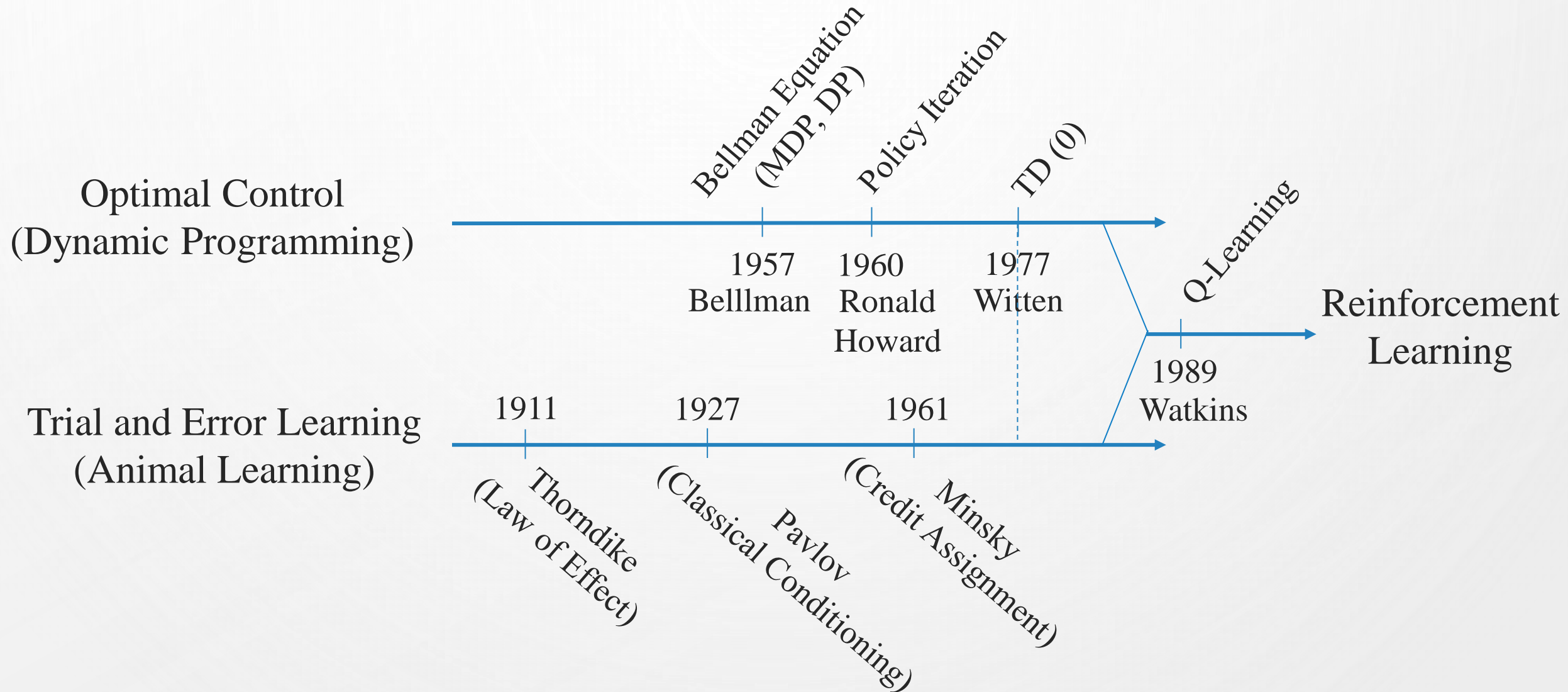
- Não existe um supervisor ou amostras rotuladas, apenas um sinal de **recompensa**.
- Feedback pode ser **atrasado**, não instantâneo.
- Influência temporal (dados são **sequenciais** e não i.i.d).
- **Interação com ambiente**: Ações tomadas pelo agente alteram o estado do ambiente e afetam dados subsequentes (exploração).

APRENDIZADO POR REFORÇO: ÁREA DO CONHECIMENTO



Fonte: UCL Course on RL by David Silver
(<https://www.davidsilver.uk/teaching/>)

APRENDIZADO POR REFORÇO: HISTÓRIA



APRENDIZADO POR REFORÇO: APLICAÇÕES

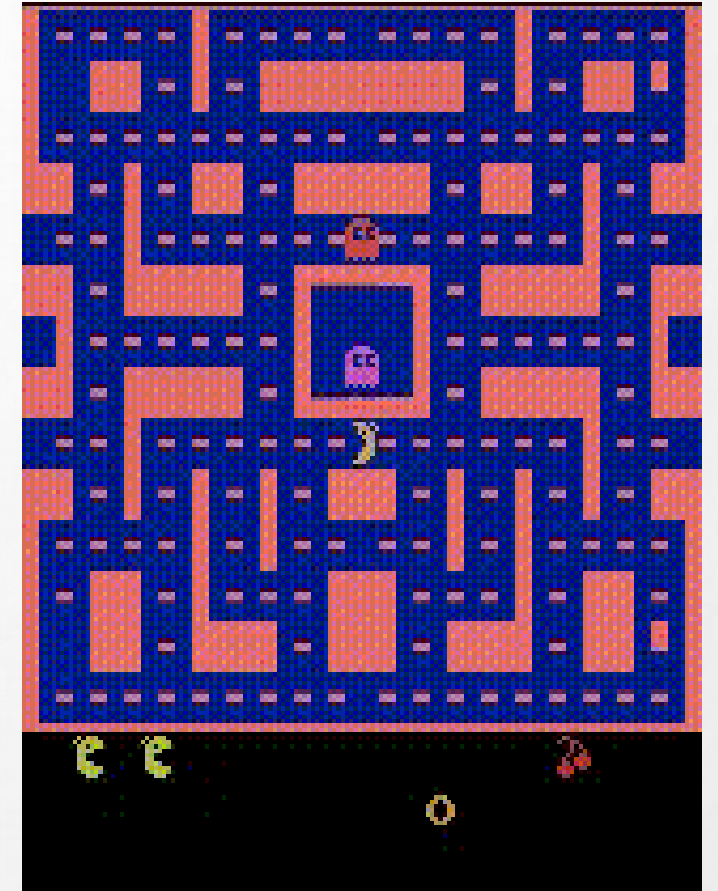
Aprendizado por Reforço: Aplicações

APRENDIZADO POR REFORÇO: APLICAÇÕES

- **Jogos Eletrônicos:** Desenvolvimento de IA capaz de jogar diversos videogames.

Exemplo: PacMan (atari)

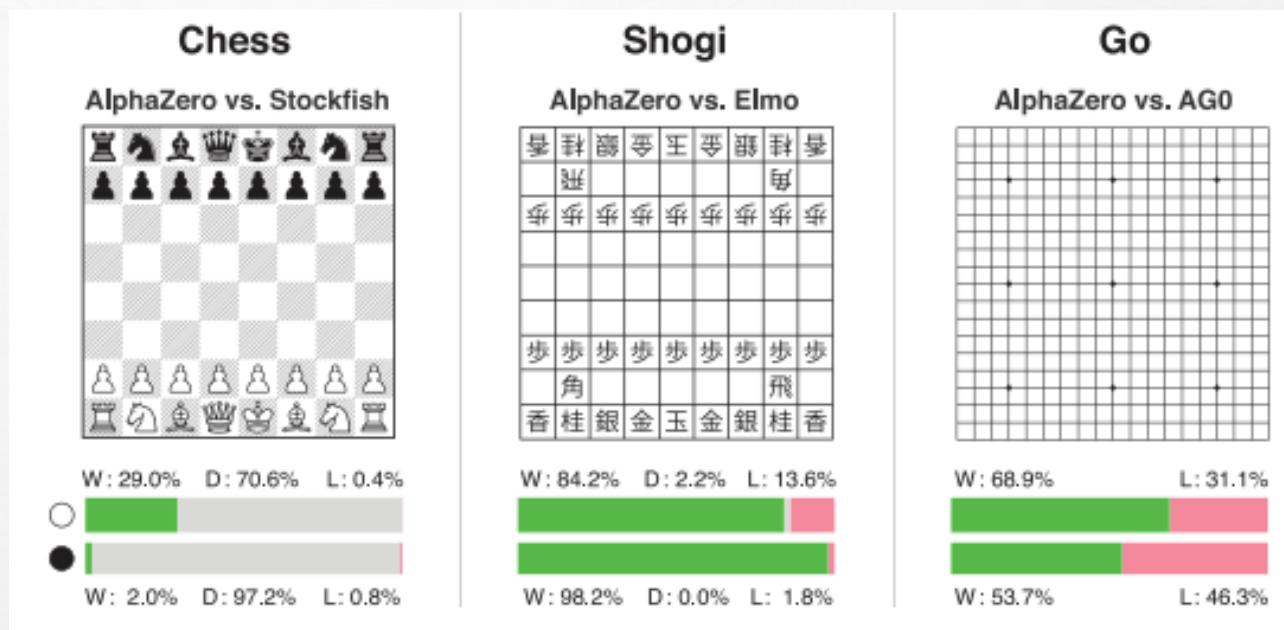
- **Estado:** Posição de personagem, inimigos e moedas.
- **Ação:** Movimento a ser efetuado (para cima, baixo, esquerda, direita).
- **Recompensa:** Pontuação.



APRENDIZADO POR REFORÇO: APLICAÇÕES

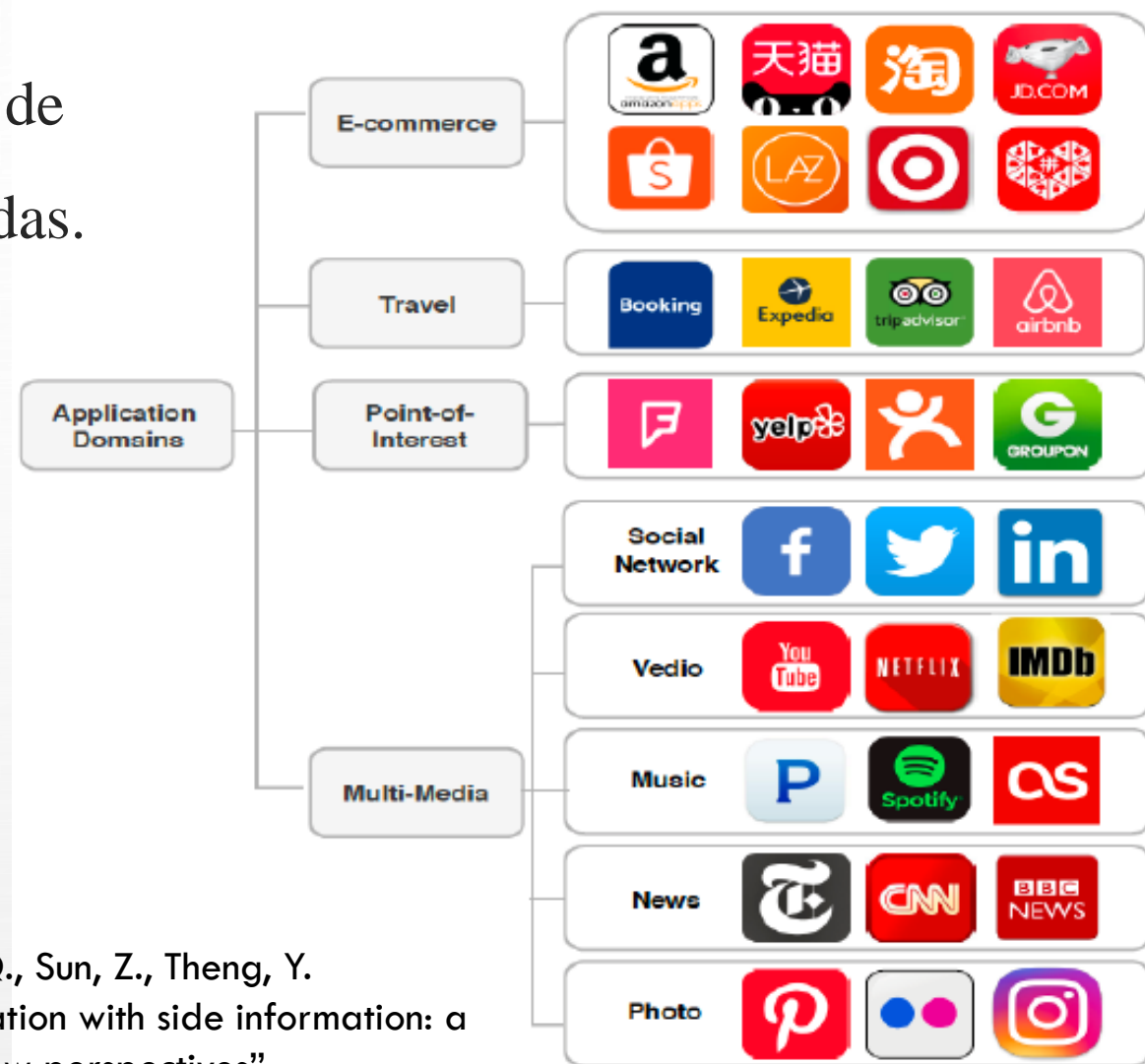
- **Jogos de Tabuleiro:** Desenvolvimento de IA capaz de jogar com humanos.
 - **Estado:** Posições de peças.
 - **Ação:** Movimento a ser efetuado.
 - **Recompensa:** Vitória/Derrota/Empate.

Fonte: Silver, D., "Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm", 2017



APRENDIZADO POR REFORÇO: APLICAÇÕES

- **Sistemas de Recomendação:** Recomendação de produtos para cada usuário em um site de vendas.
 - **Estado:** Histórico de compras e produtos visualizados pelo usuário.
 - **Ação:** Produtos a serem recomendados.
 - **Recompensa:** Definida em função de compra/visualização do produto.



Fonte: Guo, Q., Sun, Z., Theng, Y.
 “Recommendation with side information: a survey and new perspectives”

APRENDIZADO POR REFORÇO: APLICAÇÕES

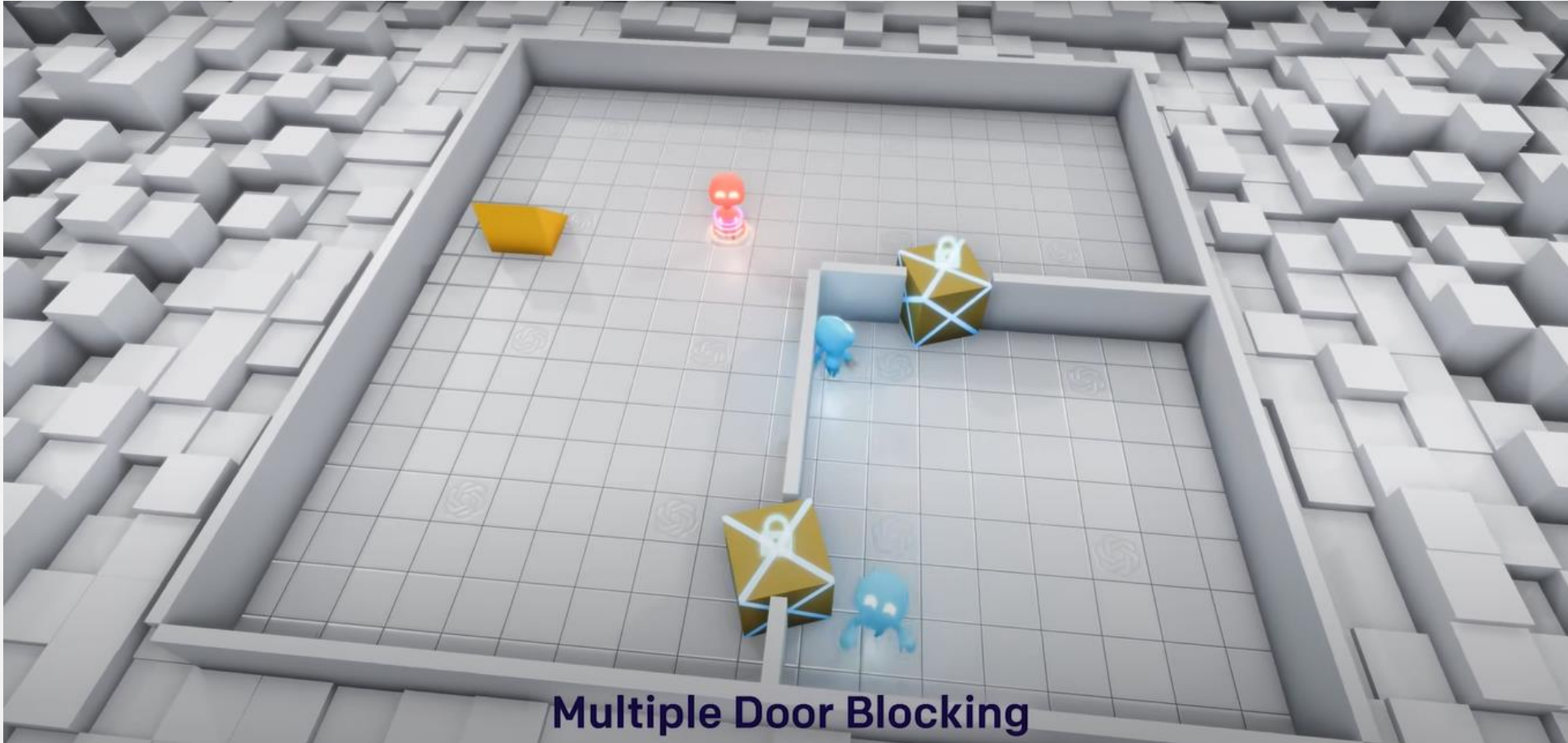
AlphaStar



<https://www.youtube.com/watch?v=UuhECwm31dM>

APRENDIZADO POR REFORÇO: APLICAÇÕES

OpenAI Hide and Seek



<https://www.youtube.com/watch?v=kopolzvh5jY&>

APRENDIZADO POR REFORÇO: APLICAÇÕES

Autonomous Driving



<https://www.youtube.com/watch?v=pUhckFVXN2A>

Learning Robust Control Policies for End-to-End Autonomous Driving From Data-Driven Simulation

Alexander Amini¹, Igor Gilitschenski², Jacob Phillips, Julia Moseyko,
Rohan Banerjee³, Sertac Karaman⁴, and Daniela Rus¹

Abstract—In this work, we present a data-driven simulation and training engine capable of learning end-to-end autonomous vehicle control policies using only sparse rewards. By leveraging real, human-collected trajectories through an environment, we render novel training data that allows virtual agents to drive along a continuum of new local trajectories consistent with the road appearance and semantics, each with a different view of the scene. We demonstrate the ability of policies learned within our simulator to generalize and navigate in previously unseen real-world roads, without access to any human control labels during training. Our results validate the learned policy onboard a full-scale autonomous vehicle, including in previously unencountered scenarios, such as new roads and novel, complex, near-crash situations. Our methods are scalable, leverage reinforcement learning, and apply broadly to situations requiring effective perception and robust operation in the physical world.

Index Terms—Deep learning in robotics and automation, autonomous agents, real world reinforcement learning, data-driven simulation.

I. INTRODUCTION

END-TO-END (i.e., perception-to-control) trained neural networks for autonomous vehicles have shown a great promise for lane stable driving [1]–[3]. However, they lack methods to learn robust models at scale and require vast amounts of training data that are time consuming and expensive to collect. Learned end-to-end driving policies and modular perception components in a driving pipeline require capturing training data from all necessary edge cases, such as recovery from off-orientation positions or even near collisions. This is not only prohibitively expensive, but also potentially dangerous [4]. Training and evaluating robotic controllers in simulation [5]–[7]

Manuscript received September 10, 2019; accepted December 10, 2019. Date of publication January 13, 2020; date of current version January 10, 2020. This letter was recommended for publication by Associate Editor E. E. Akay and Editor T. Adar after several rounds of review. This work was supported in part by National Science Foundation (NSF) Grant IRI-1710000, Toyota Research Institute (TRI), and in part by NVIDIA Corporation. (Corresponding author: Alexander Amini.)

¹A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, and D. Rus are with the Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America (e-mail: amini@mit.edu, igilitsch@mit.edu, jphillips@mit.edu, jmosseyko@mit.edu, rbanerjee@mit.edu, drus@mit.edu).

²S. Karaman is with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA 02139, United States of America (e-mail: sertac@mit.edu).

This letter has supplementary downloadable material available at <https://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LRA.2020.2966414

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see <http://creativecommons.org/licenses/by/4.0/>

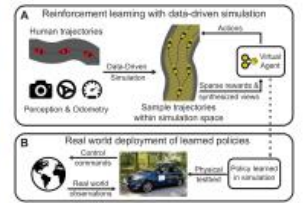


Fig. 1. Training and deployment of policies from data-driven simulation: From a single human collected trajectory our data-driven simulation (VISTA) synthesizes a space of new possible trajectories for learning virtual agent control policies (A). Perceiving photorealistic of the real world allows the virtual agent to move beyond imitation learning and instead explore the space using reinforcement learning with only sparse rewards. Learned policies not only transfer directly to the real world (B), but also outperform state-of-the-art end-to-end methods trained using imitation learning.

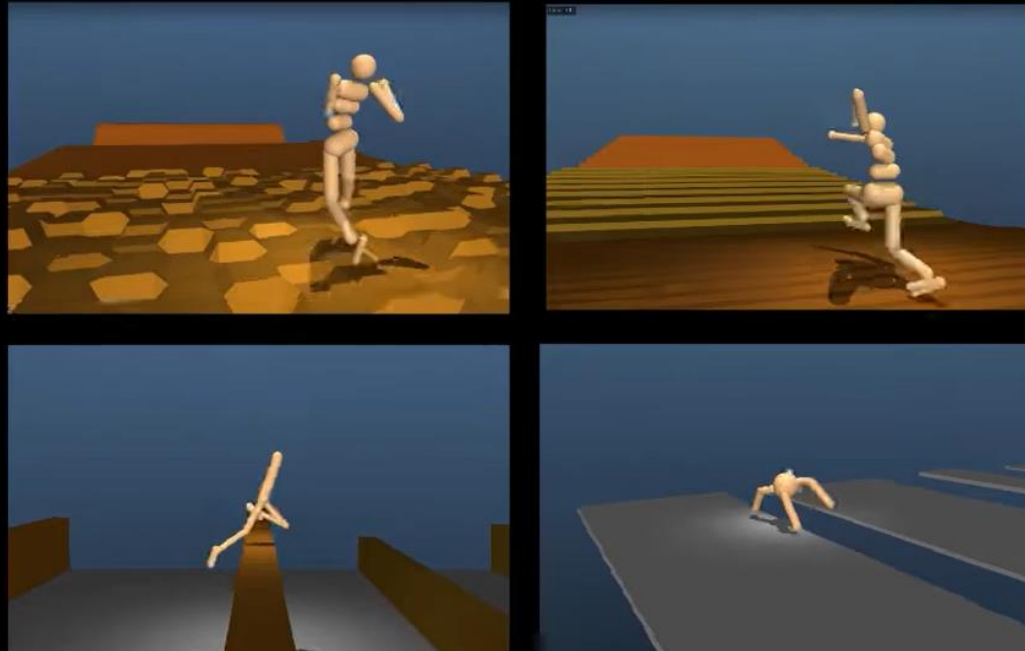
has emerged as a potential solution to the need for more data and increased robustness to novel situations, while also avoiding the time, cost, and safety issues of current methods. However, transferring policies learned in simulation into the real-world still remains an open research challenge. In this letter, we present an end-to-end simulation and training engine capable of training real-world reinforcement learning (RL) agents entirely in simulation, without any prior knowledge of human driving or post-training fine-tuning. We demonstrate trained models can then be deployed directly in the real world, on roads and environments not encountered in training. Our engine, termed **VISTA: Virtual Image Synthesis and Transformation for Autonomous**, synthesizes a continuum of driving trajectories that are photorealistic and semantically faithful to their respective real world driving conditions (Fig. 1), from a small dataset of human collected driving trajectories. **VISTA** allows a virtual agent to not only observe a stream of sensory data from stable driving (i.e., human collected driving data), but also from a simulated band of new observations from off-orientations on the road. Given visual observations of the environment (i.e., camera images), our system learns a lane-stable control policy

<https://ieeexplore.ieee.org/document/8957584#full-text-section>

APRENDIZADO POR REFORÇO: APLICAÇÕES

DeepMind dm_control

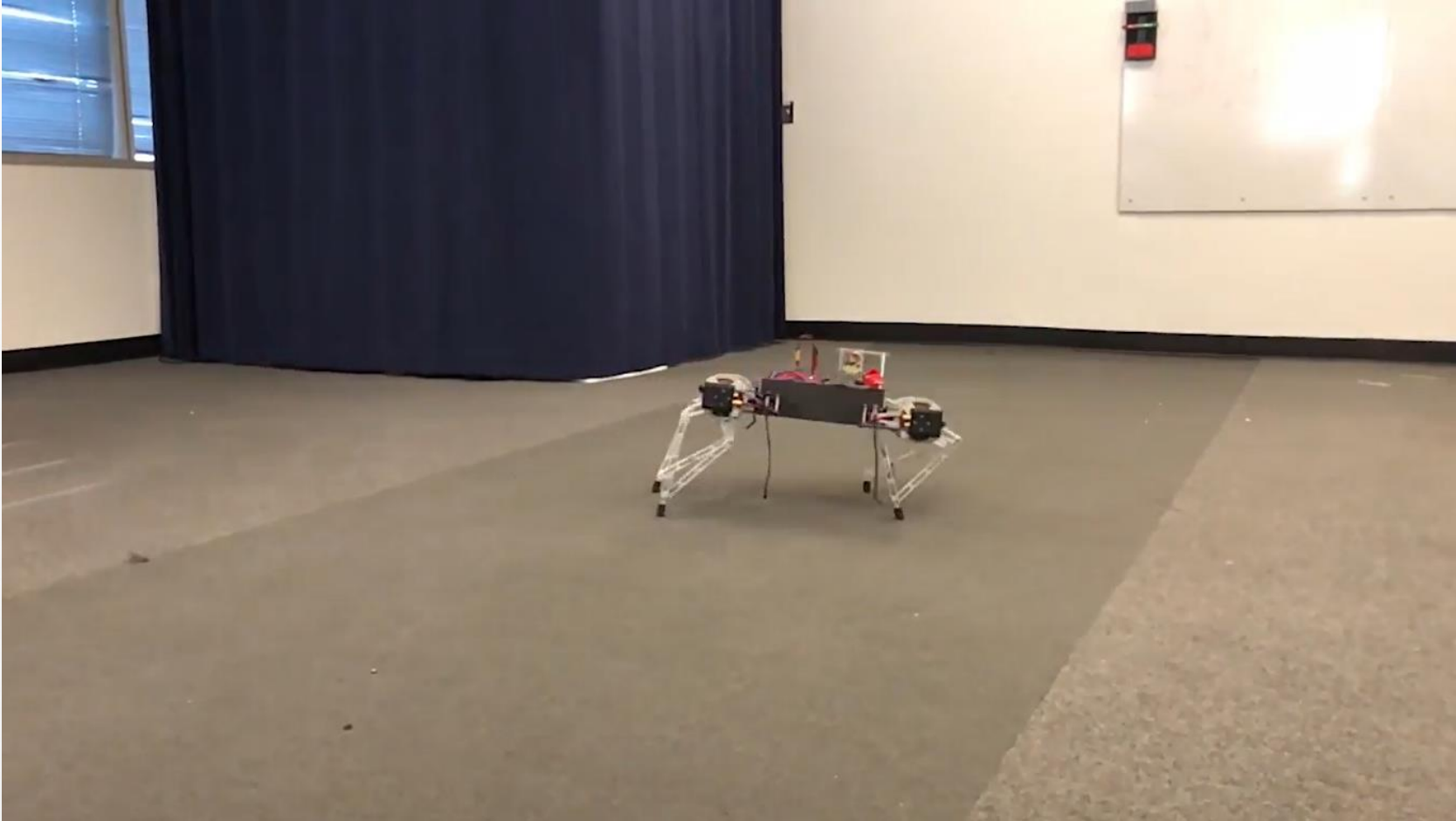
Emergence of Locomotion Behaviours in Rich Environments
(Heess et al., 2017)



<https://www.youtube.com/watch?v=CMjoiU482Jk>

APRENDIZADO POR REFORÇO: APLICAÇÕES

DeepMind dm_control



<https://www.youtube.com/watch?v=IUZUr7jxoqM>

APRENDIZADO POR REFORÇO: PRINCIPAIS CONCEITOS

Aprendizado por Reforço:
Principais Conceitos

RECOMPENSAS

- Uma **recompensa** R_t é um sinal escalar de feedback.
- R_t indica a qualidade da ação tomada pelo agente no instante t .
- O objetivo do agente é maximizar as recompensas ao longo do tempo.

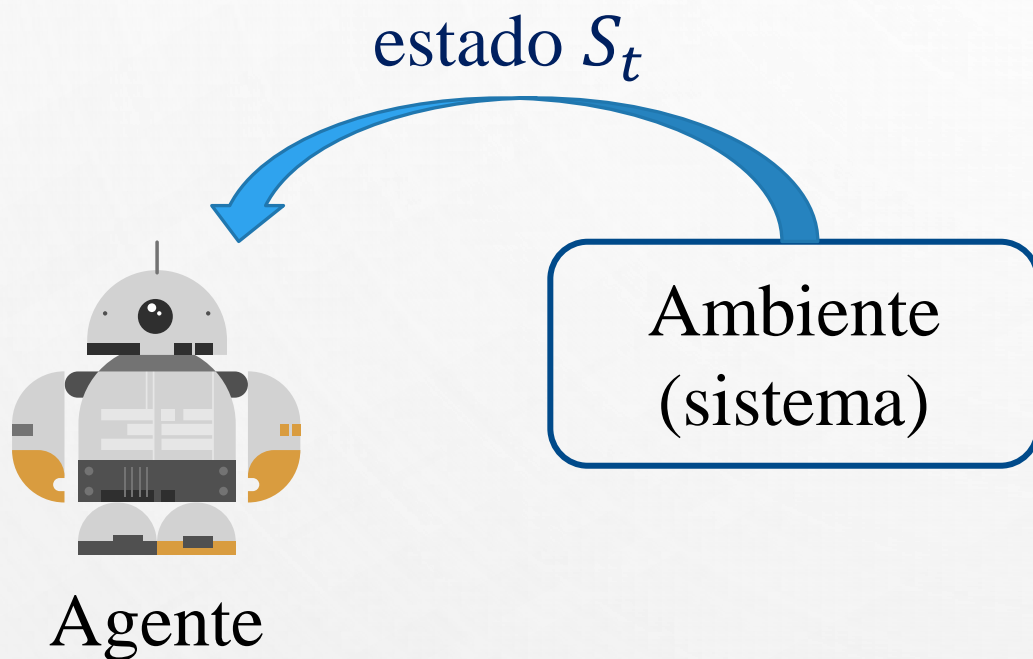
Reward Hypothesis: Todo objetivo pode ser descrito pela maximização de recompensas acumuladas.

Dada uma tarefa, é possível definir uma função recompensa $r(\mathbf{s}, \mathbf{a})$, de modo que o comportamento ótimo que soluciona a tarefa é aquele que maximiza o valor esperado dessa recompensa ao longo do tempo.

TOMADA DE DECISÕES SEQUENCIAL

A cada instante de tempo o agente:

- Observa o estado do ambiente.

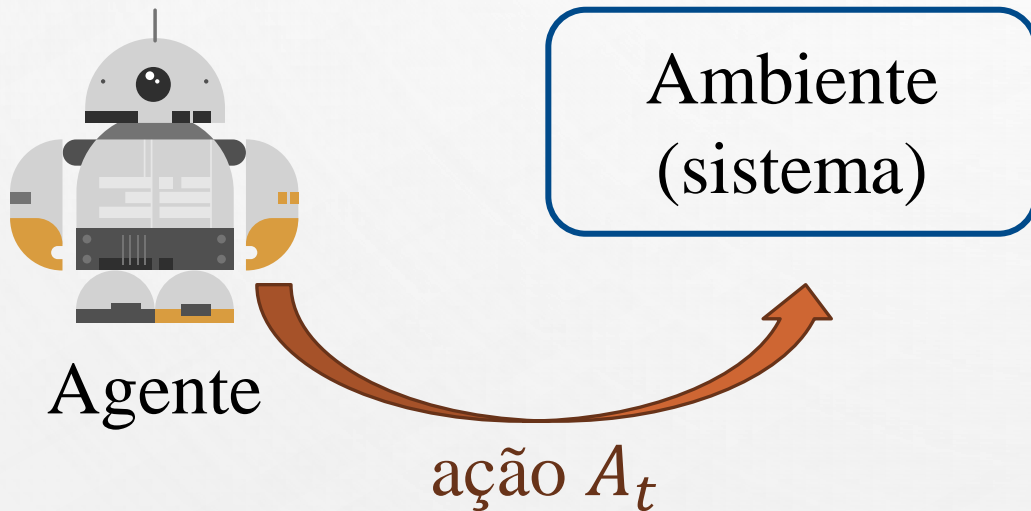


TOMADA DE DECISÕES SEQUENCIAL

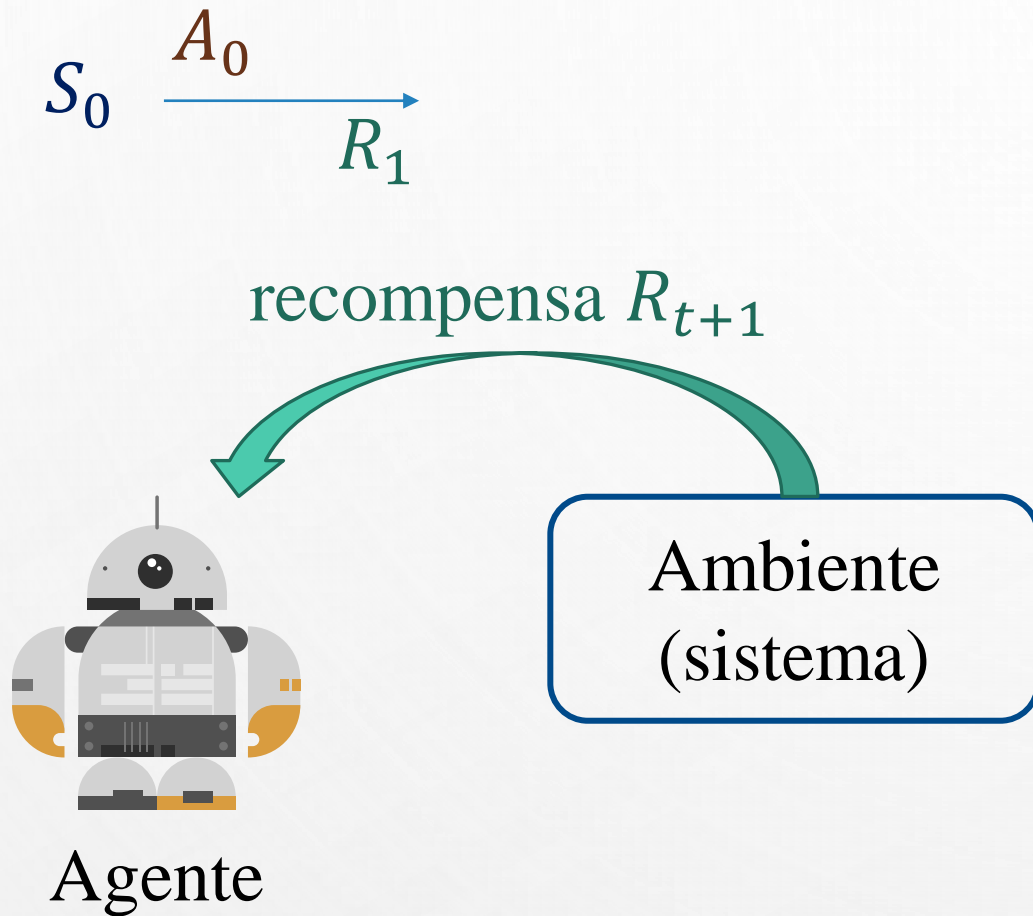
$S_0 \xrightarrow{A_0}$

A cada instante de tempo o agente:

- Observa o estado do ambiente.
- Escolhe e executa uma ação.



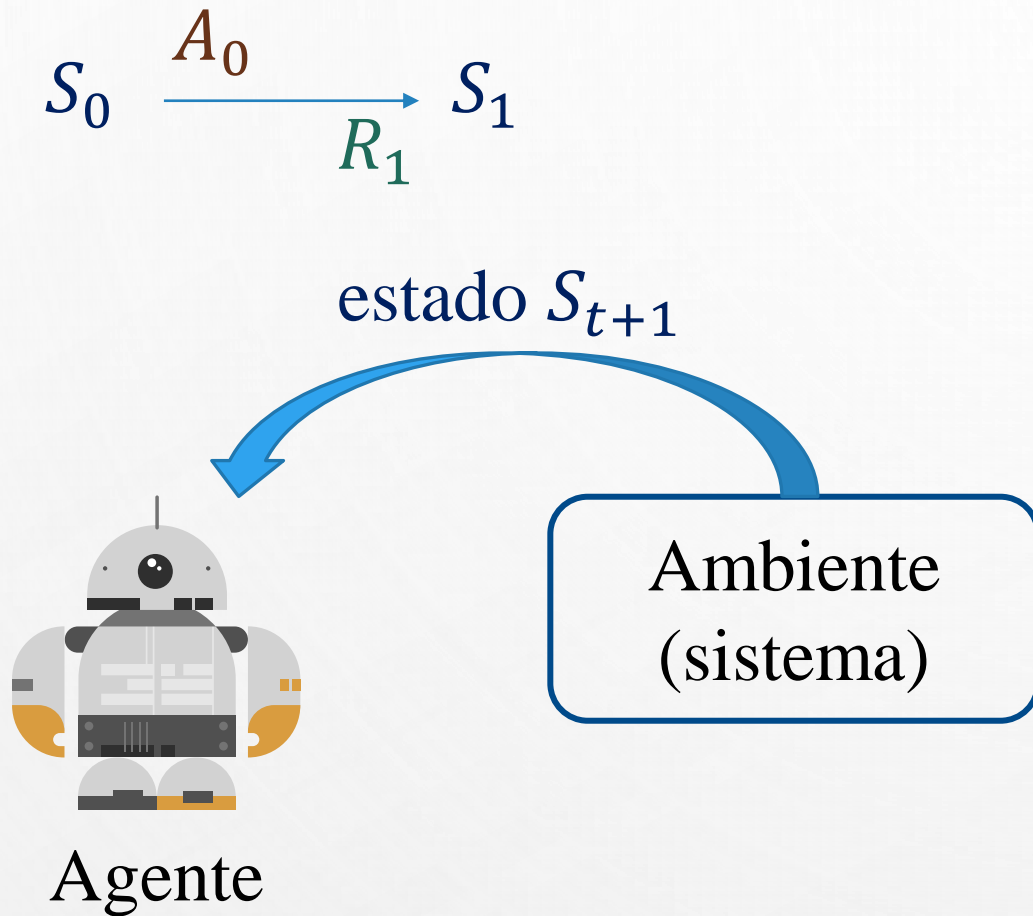
TOMADA DE DECISÕES SEQUENCIAL



A cada instante de tempo o agente:

- Observa o estado do ambiente.
- Escolhe e executa uma ação.
- Recebe uma recompensa imediata.

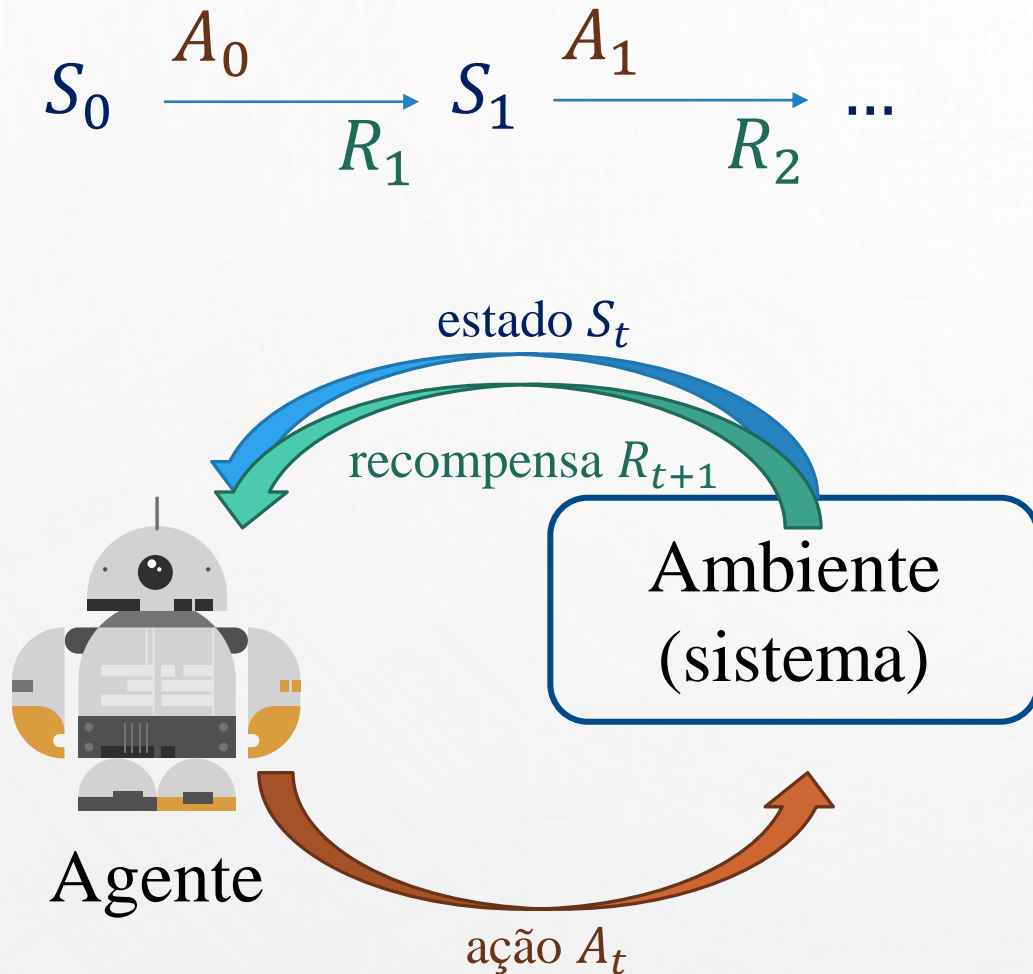
TOMADA DE DECISÕES SEQUENCIAL



A cada instante de tempo o agente:

- Observa o estado do ambiente.
- Escolhe e executa uma ação.
- Recebe uma recompensa imediata.
- O sistema evolui para um novo estado.

TOMADA DE DECISÕES SEQUENCIAL



A cada instante de tempo o agente:

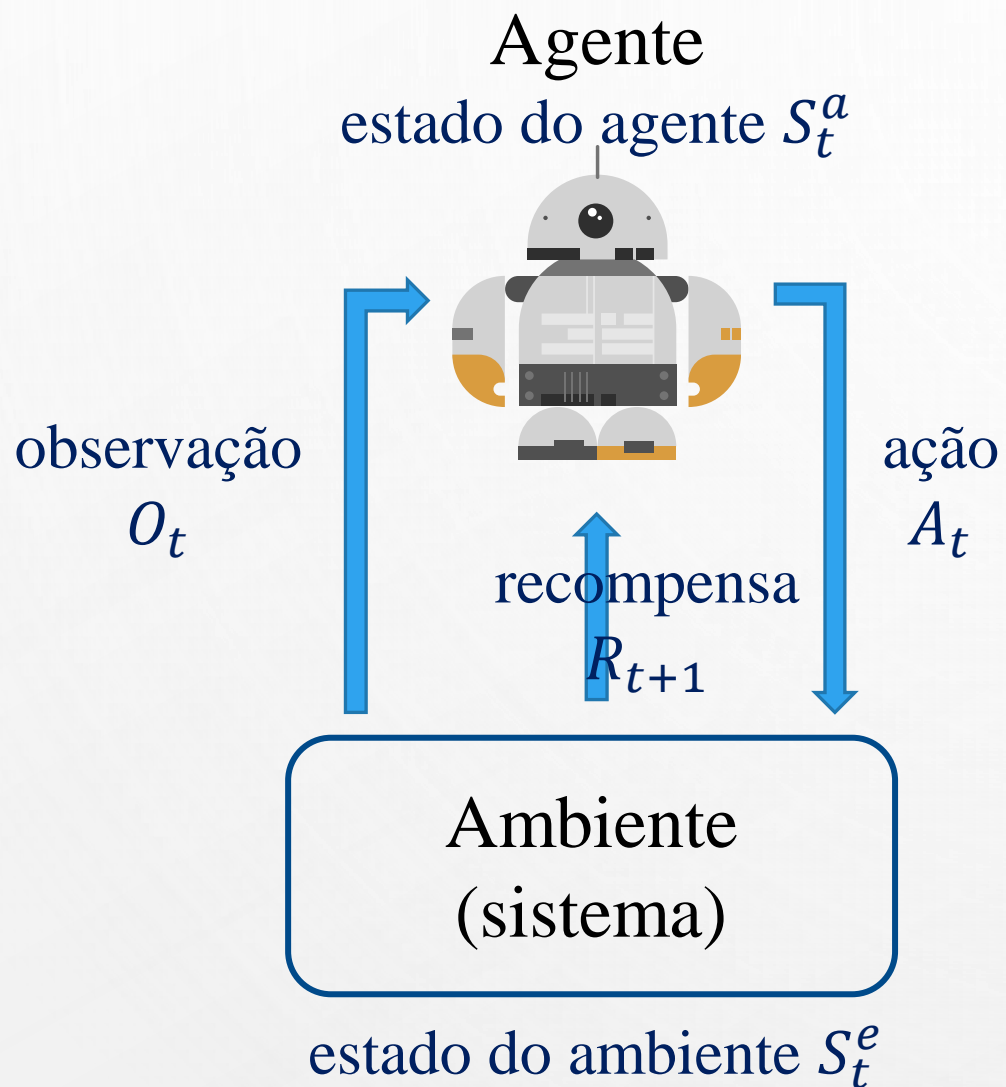
- Observa o estado do ambiente.
- Escolhe e executa uma ação.
- Recebe uma recompensa imediata.
- O sistema evolui para um novo estado.

O processo é então repetido.

Tempo discreto: Decisões são tomadas somente em épocas de decisão

$$t \in \{0, 1, \dots, N\}$$

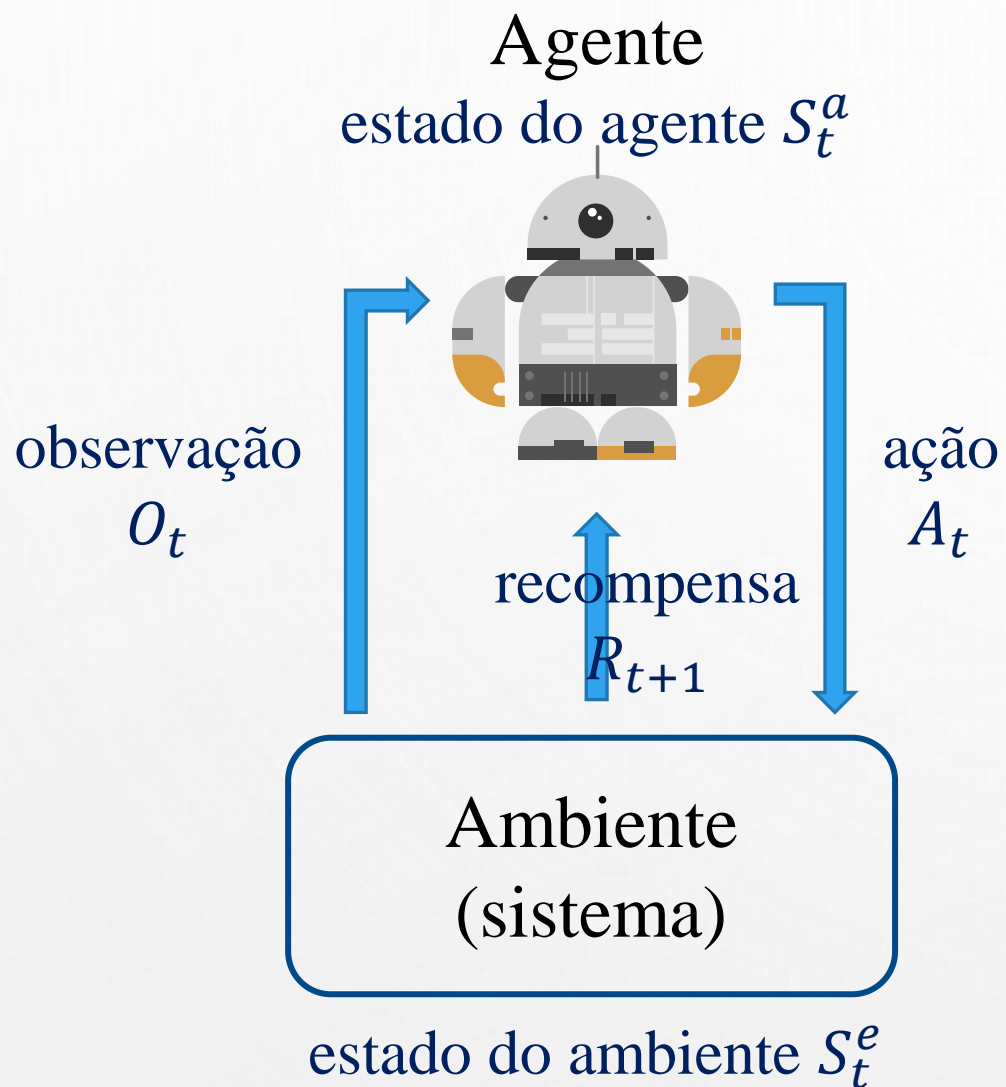
AGENTE E AMBIENTE



- A experiência do agente é dada por uma história, ou sequência de observações, ações e recompensas:

$$H_t = O_0, A_0, R_1, \dots, O_{t-1}, A_{t-1}, R_t$$

AGENTE E AMBIENTE

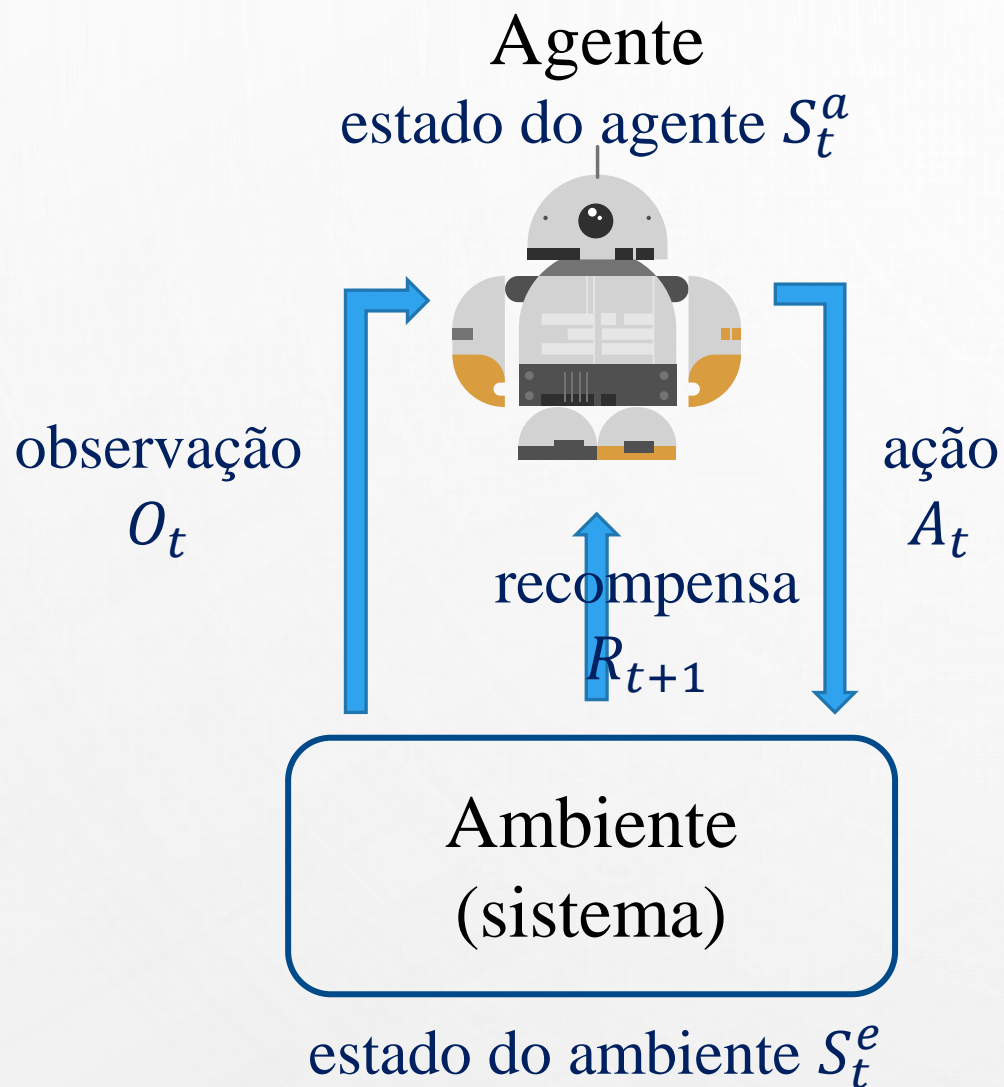


- A experiência do agente é dada por uma história, ou sequência de observações, ações e recompensas:

$$H_t = O_0, A_0, R_1, \dots, O_{t-1}, A_{t-1}, R_t$$

- O **estado do ambiente** S_t^e não é necessariamente conhecido pelo agente.

AGENTE E AMBIENTE



- A experiência do agente é dada por uma história, ou sequência de observações, ações e recompensas:

$$H_t = O_0, A_0, R_1, \dots, O_{t-1}, A_{t-1}, R_t$$

- O **estado do ambiente S_t^e** não é necessariamente conhecido pelo agente.
- O **estado do agente S_t^a** é sua representação interna do estado do ambiente e é uma função da história:

$$S_t^a = f(H_t)$$

ESTADOS DE MARKOV

Um estado de **Markov** S_t contém toda a informação útil da história H_t :

- Um estado S_t é de Markov se, e somente se, satisfaz a **propriedade de Markov**:

$$\mathbb{P}(S_{t+1}|S_t) = \mathbb{P}(S_{t+1}|S_0, \dots, S_t)$$

- Ou seja, o estado futuro independe de estados passados dado o estado atual.
- O estado S_t é uma estatística suficiente do futuro.
- Um agente ótimo pode tomar decisões com base apenas em S_t , sem a necessidade de conhecer como o estado S_t foi alcançado.

O estado do ambiente S_t^e é de Markov.

COMO DETERMINAR O ESTADO?

Observações:

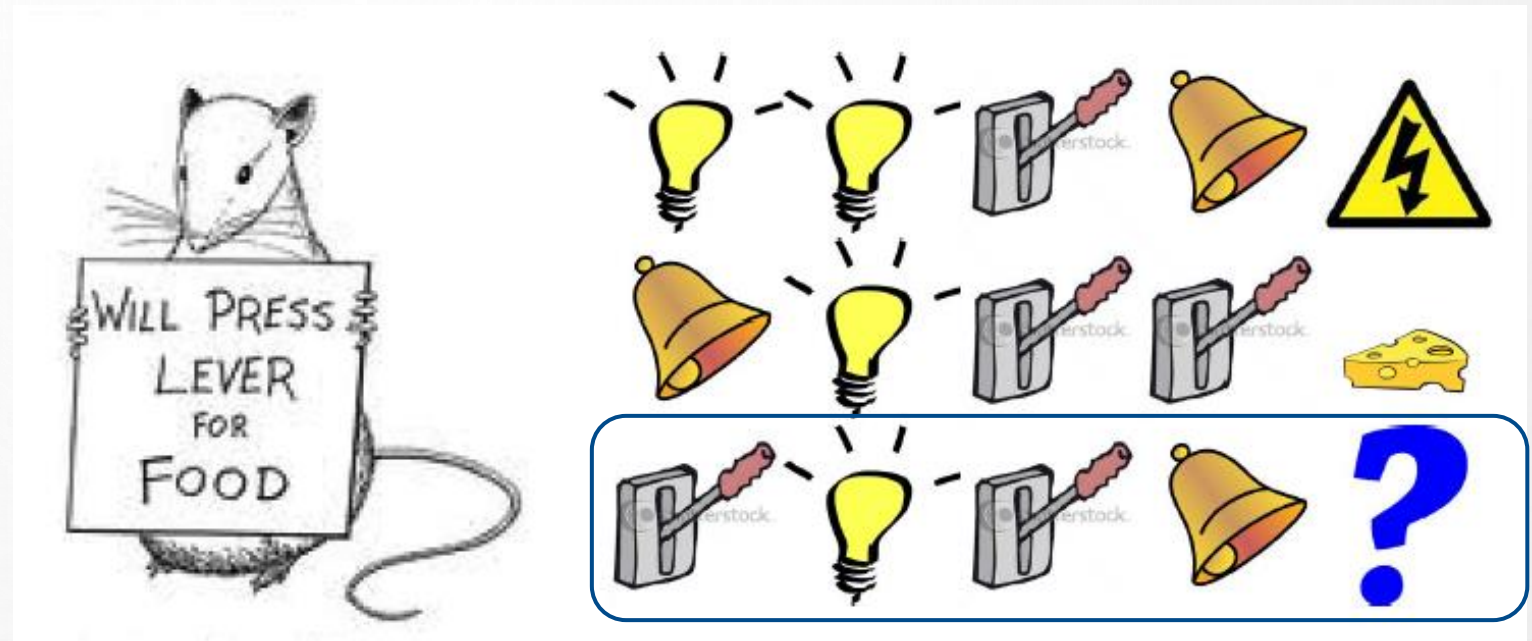
- O_A : Lâmpada
- O_B : Sino

Ações:

- A_A : Não fazer nada
- A_B : Puxar alavanca

Recompensas:

- R_A : Choque ($R_A = -10$)
- R_B : Queijo ($R_B = 10$)
- R_C : Nada ($R_C = 0$)



Fonte: UCL Course on RL by David Silver (<https://www.davidsilver.uk/teaching/>)

O que espera-se que aconteça na última sequência?

COMO DETERMINAR O ESTADO?

Observações:

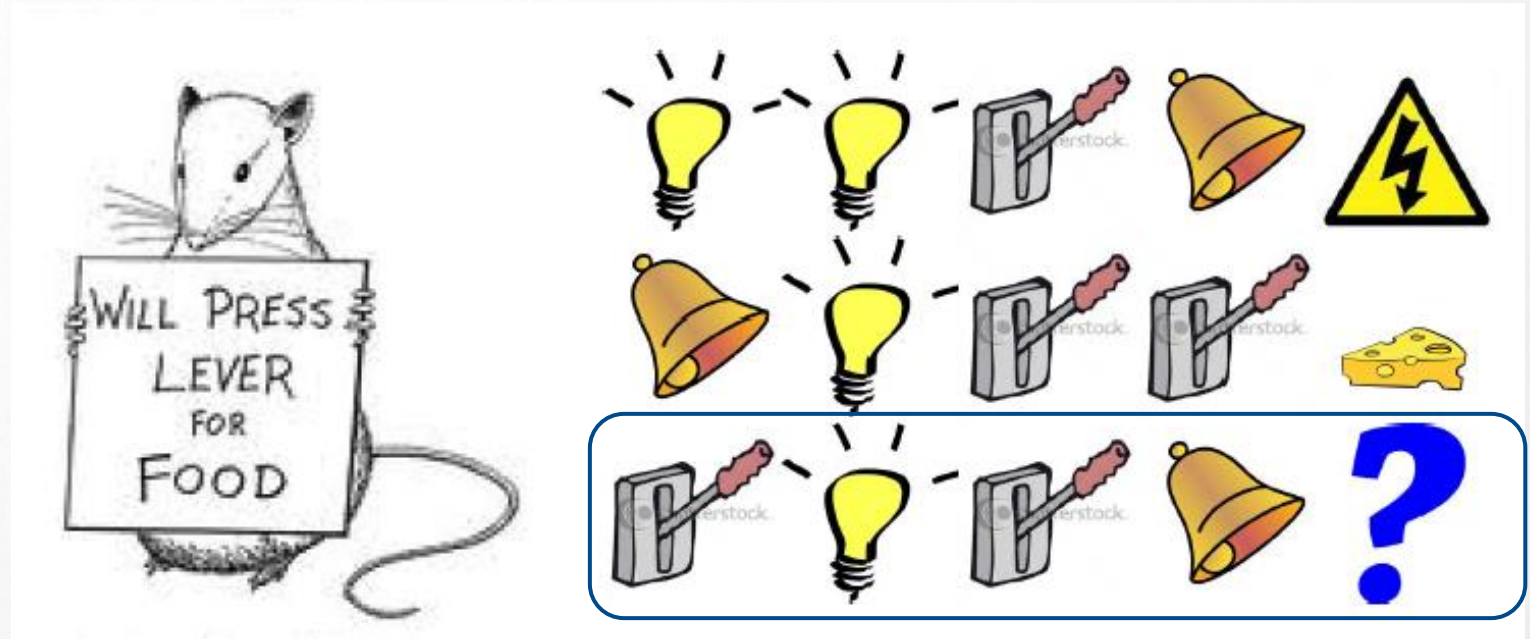
- O_A : Lâmpada
- O_B : Sino

Ações:

- A_A : Não fazer nada
- A_B : Puxar alavanca

Recompensas:

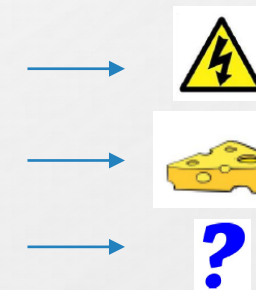
- R_A : Choque ($R_A = -10$)
- R_B : Queijo ($R_B = 10$)
- R_C : Nada ($R_C = 0$)



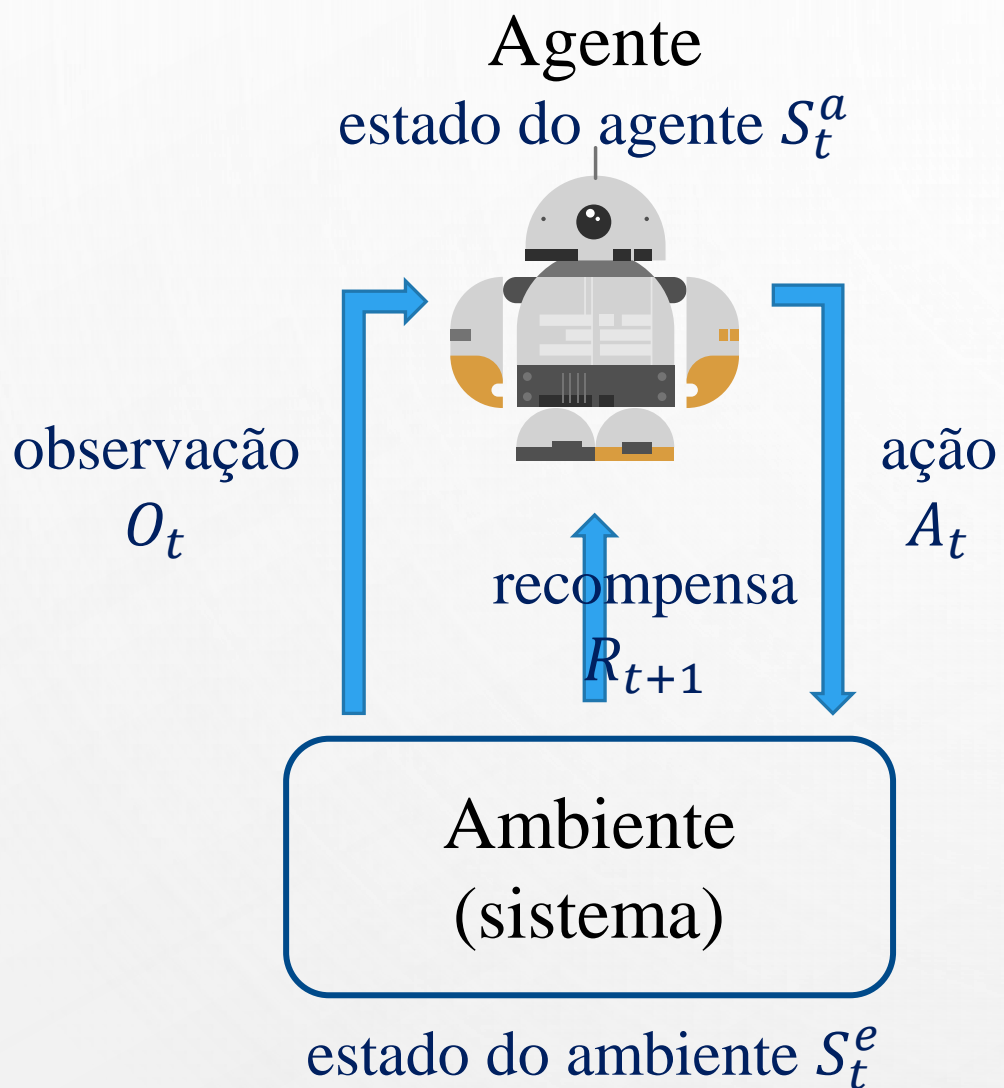
Fonte: UCL Course on RL by David Silver (<https://www.davidsilver.uk/teaching/>)

O que espera-se que aconteça na última sequência?

- S_t = última observação?
- S_t = Número de lâmpadas, sinos e alavancas?
- S_t = História completa?



OBSERVABILIDADE



Um ambiente é **completamente observável** se:

$$O_t = S_t^a = S_t^e$$

- Estado do agente = Estado do ambiente = Estado de Markov.
- Constitui um **Processo de Decisão de Markov (MDP)**.

Um ambiente é **parcialmente observável** se :

$$S_t^a \neq S_t^e$$

- Agente observa ambiente indiretamente (ausência de informação completa).
- Constitui um **Processo de Decisão de Markov Parcialmente Observável (POMDP)**.

COMPONENTES DE UM AGENTE DE RL

Um **agente** de Aprendizado por Reforço deve incluir um ou mais dos seguintes componentes:

- **Política de Ações:** Função que determina ação a ser tomada em função do estado.
- **Função Valor:** Função que determina a qualidade de um estado ou de um par estado/ação.
- **Modelo do sistema:** Representação do ambiente interna ao agente.

POLÍTICA DE AÇÕES

Uma **Política de Ações** é o modelo do comportamento do agente:

- Seja \mathcal{S} um espaço de estados e \mathcal{A} um espaço de ações, a política de ações é um mapa de $\mathcal{S} \rightarrow \mathcal{A}$.
- O objetivo do Aprendizado por Reforço é determinar uma política de ações ótima de modo a maximizar as recompensas acumuladas ao longo do tempo.
- Uma política de ações pode ser:
 - Determinística: $\pi: \mathcal{S} \rightarrow \mathcal{A}$, tal que $\pi(s) = a$.
 - Probabilística: $\pi: \mathcal{S} \times \mathcal{A} \rightarrow [0,1]$, tal que $\pi(a|s) = \mathbb{P}(A_t = a | S_t = s)$

RETORNO G_t

- Como definir recompensas acumuladas? Soma simples de todas recompensas $\sum R_t$?

O **Retorno** G_t a partir de determinado instante de tempo t é definido como a soma descontada de recompensas obtidas a partir deste instante:

$$G_t \doteq \sum_{k=0}^{\infty} \gamma^k R_{t+k+1} = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots$$

onde $\gamma \in [0,1]$ é denominado **Fator de Desconto**.

- Recompensas mais próximas são priorizadas em relação a recompensas mais distantes.
- Matematicamente: Limita o retorno a um número finito.
- Intuitivamente: Incertezas fazem com que recompensas rápidas sejam preferíveis.

FUNÇÃO VALOR

A **Função Valor de um estado** é o valor esperado das recompensas descontadas obtidas a partir daquele estado:

$$V_{\pi}(s) \doteq \mathbb{E}_{\pi}[G_t | S_t = s, A_t \sim \pi(S_t)]$$

- É utilizada para avaliar a qualidade de um estado dada a política de ações.

A **Função Valor de um par estado/ação** é o valor esperado das recompensas descontadas obtidas depois que determinada ação foi tomada em determinado estado:

$$Q_{\pi}(s, a) \doteq \mathbb{E}_{\pi}[G_t | S_t = s, A_t = a]$$

- Pode ser utilizada para selecionar ações: “Selecionar sempre ações de maior valor”.

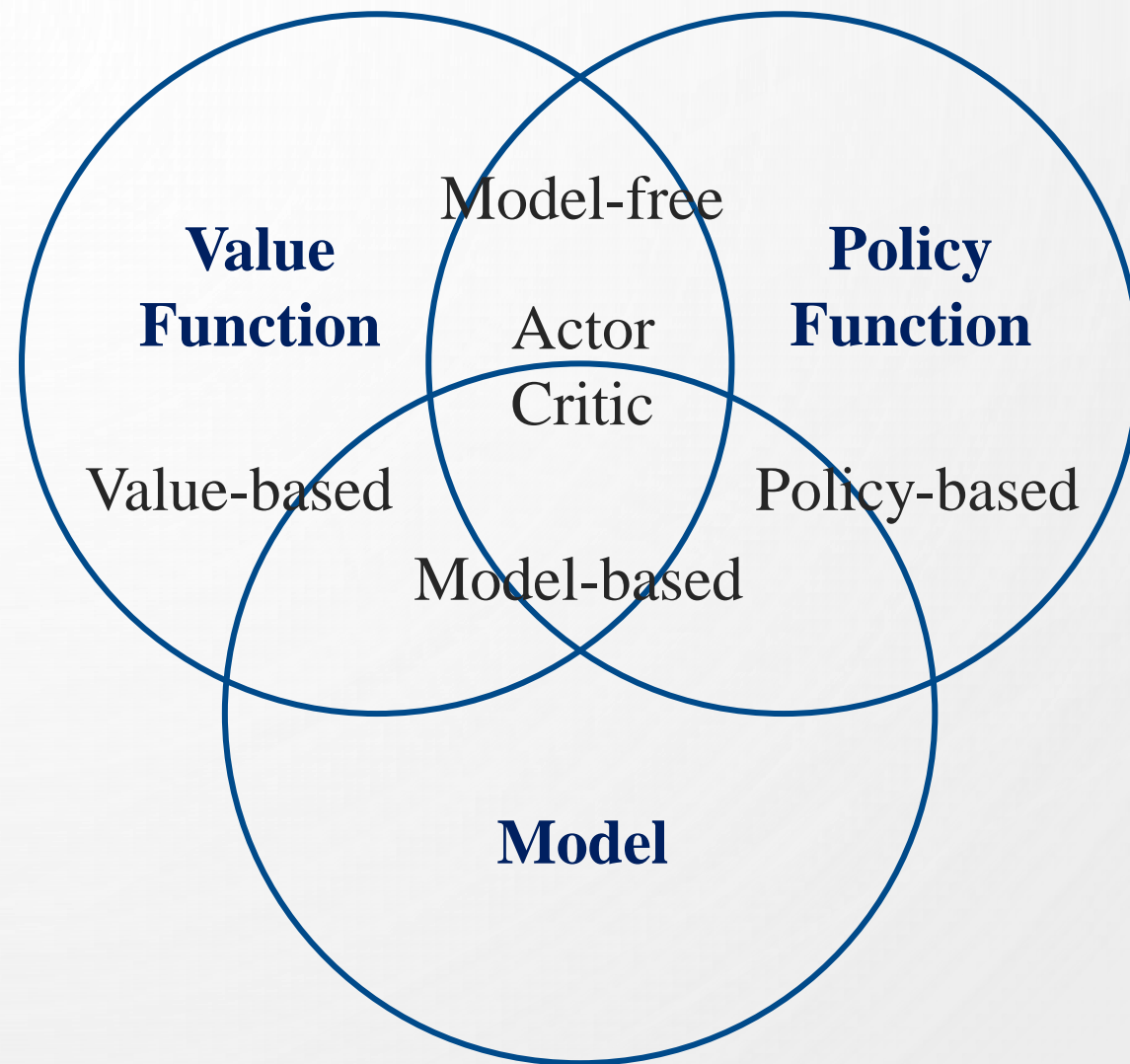
MODELO

Um **modelo** do sistema é uma representação interna ao agente do comportamento do ambiente. Um modelo $M = \{\mathcal{P}_{ss'}^a, \mathcal{R}_s^a\}$ aproxima duas funções:

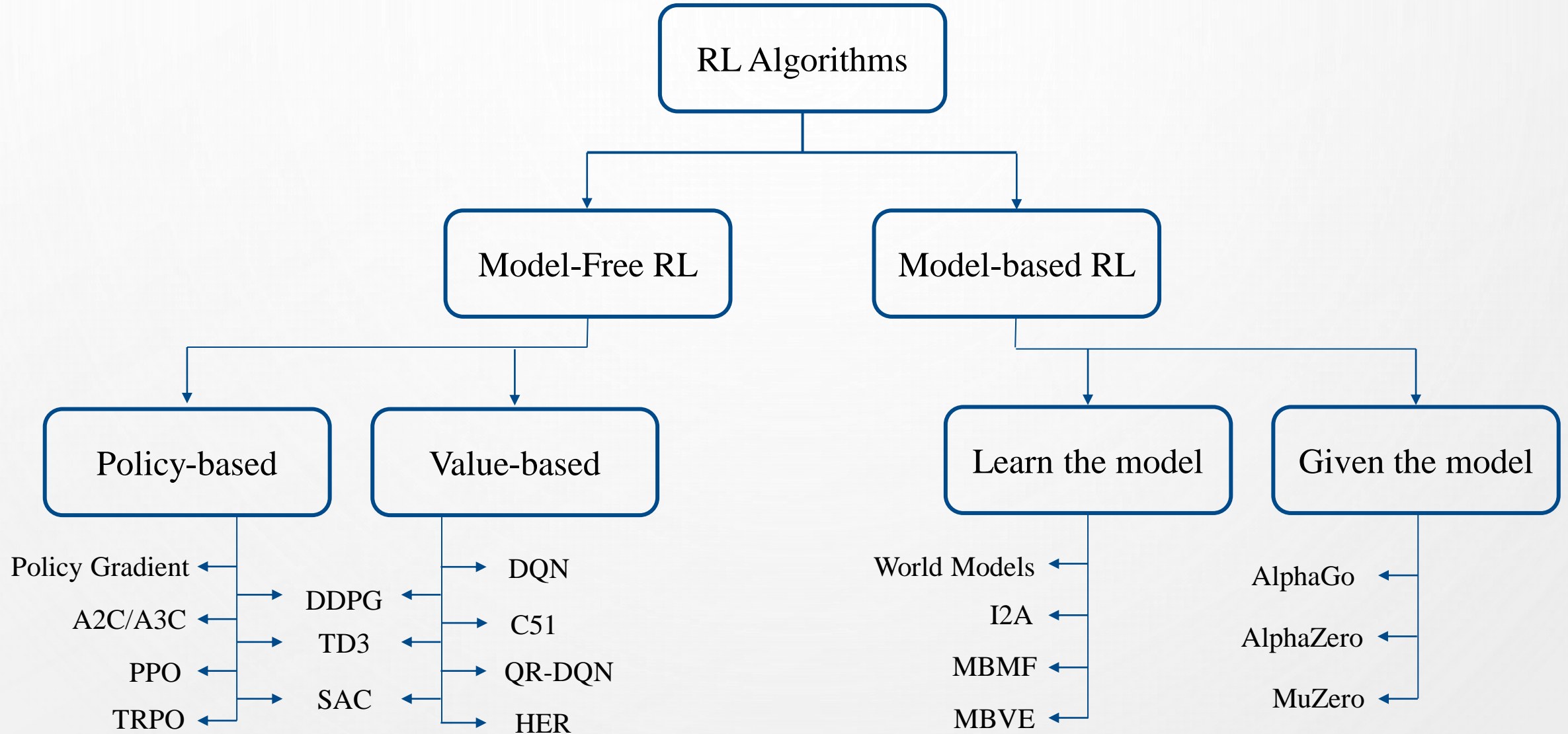
- $\mathcal{P}_{ss'}^a = \mathbb{P}(S_{t+1} = s' | S_t = s, A_t = a)$: função de transição de estados do ambiente.
- $\mathcal{R}_s^a = \mathbb{E}[R_{t+1} | S_t = s, A_t = a]$: função de recompensa imediata.

APRENDIZADO POR REFORÇO: TIPOS DE ALGORITMOS

Algoritmos de Aprendizado por Reforço podem ser categorizados de acordo com os componentes treinados e presentes no agente.

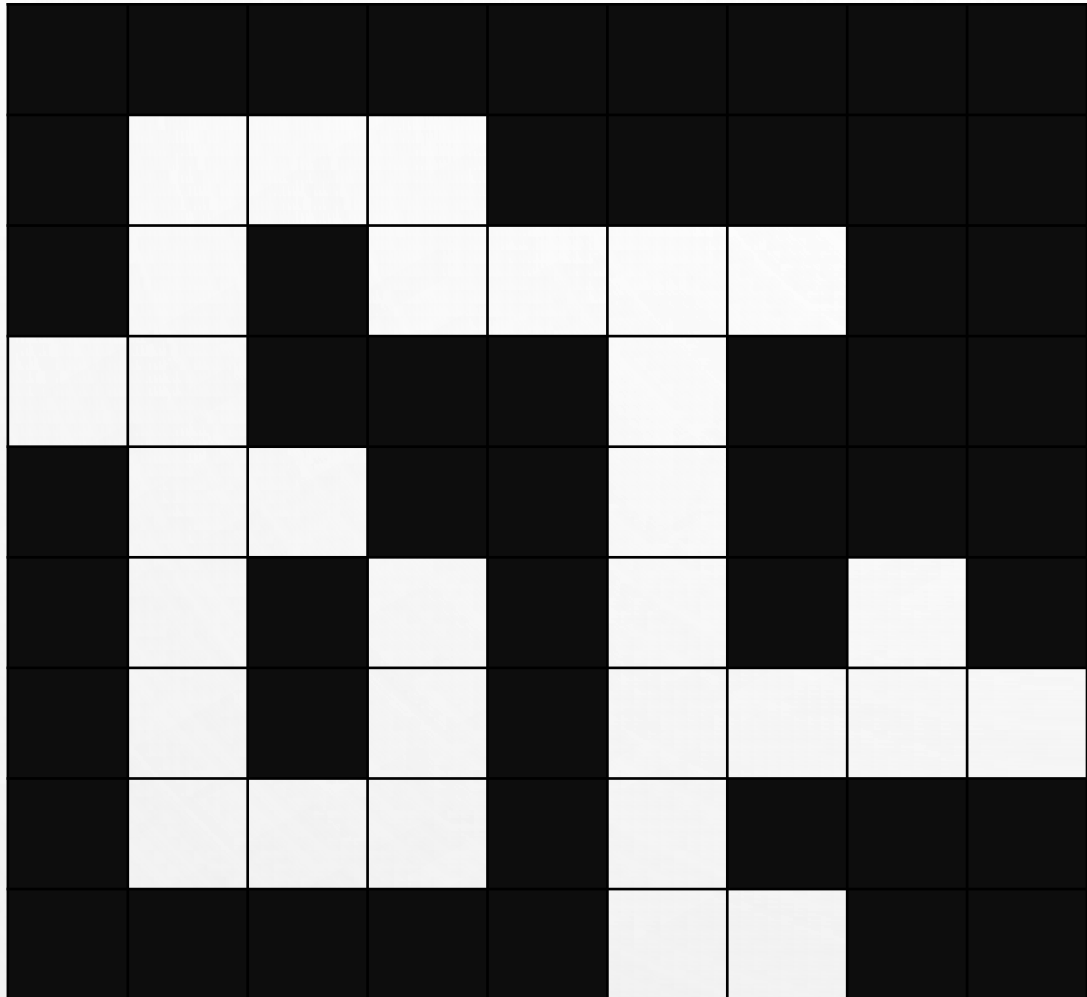


APRENDIZADO POR REFORÇO: TIPOS DE ALGORITMOS



EXEMPLO: LABIRINTO

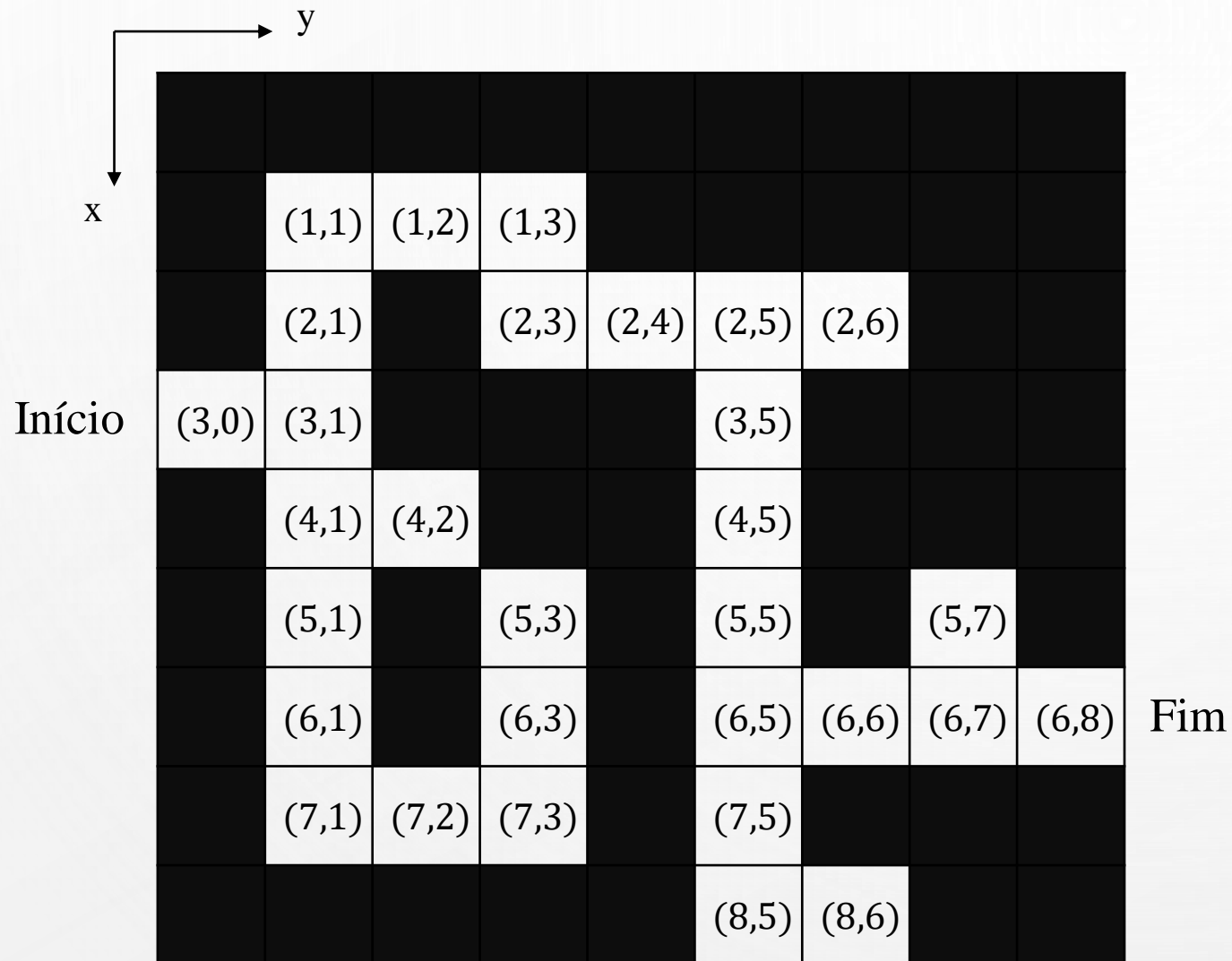
Início



Fim

Objetivo: Chegar ao final do labirinto a partir da posição inicial.

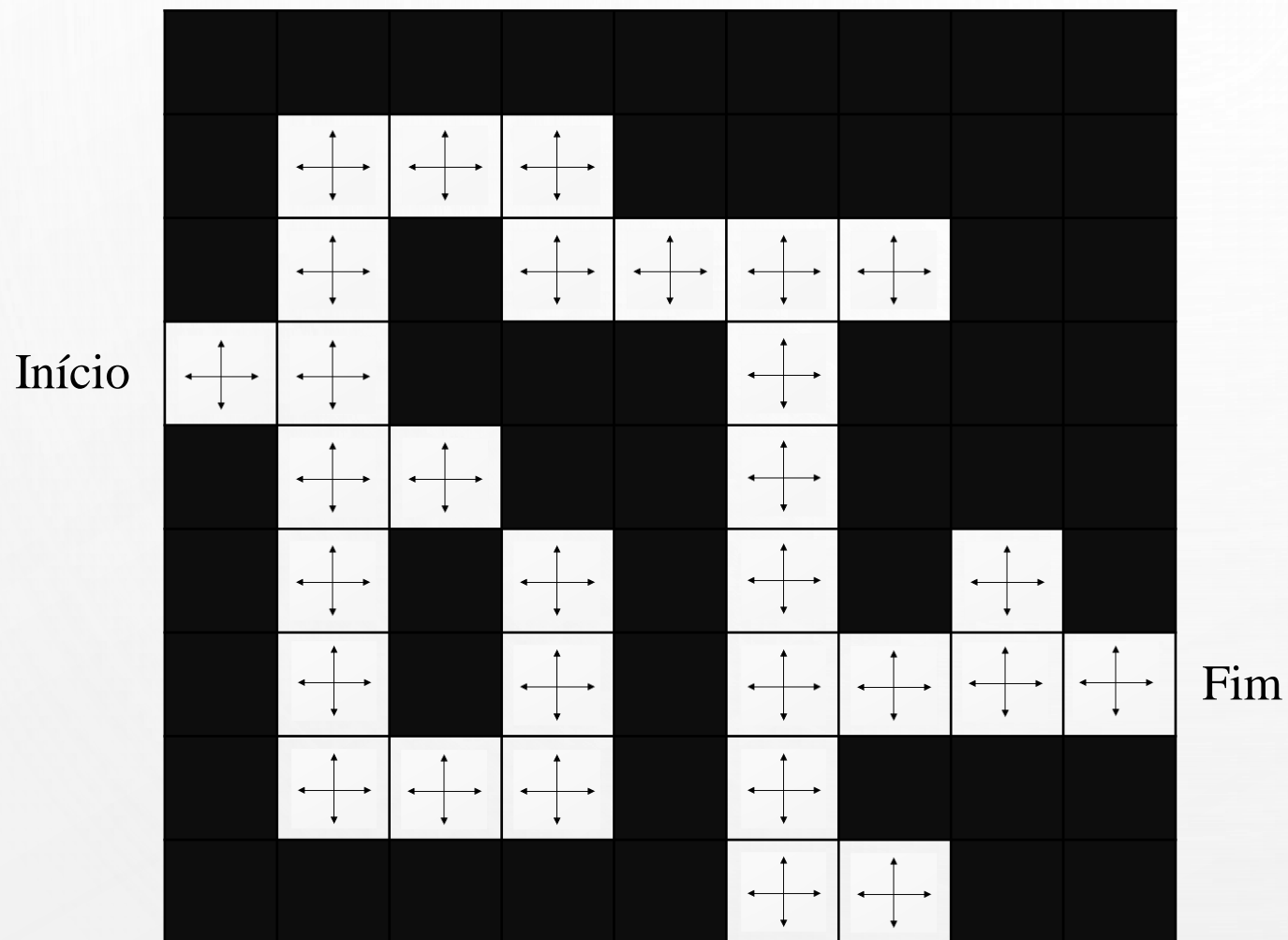
EXEMPLO: LABIRINTO



Objetivo: Chegar ao final do labirinto a partir da posição inicial.

- Estados: Posições em que o agente se encontra.

EXEMPLO: LABIRINTO



Objetivo: Chegar ao final do labirinto a partir da posição inicial.

- Estados: Posições em que o agente se encontra.
- Ações: 0, \uparrow , \downarrow , \leftarrow , \rightarrow

EXEMPLO: LABIRINTO

	-1	-1	-1						
	-1		-1	-1	-1	-1			
Início	-1	-1				-1			
	-1	-1				-1			
	-1		-1			-1		-1	
	-1		-1			-1	-1	-1	0
	-1	-1	-1			-1			
						-1	-1		

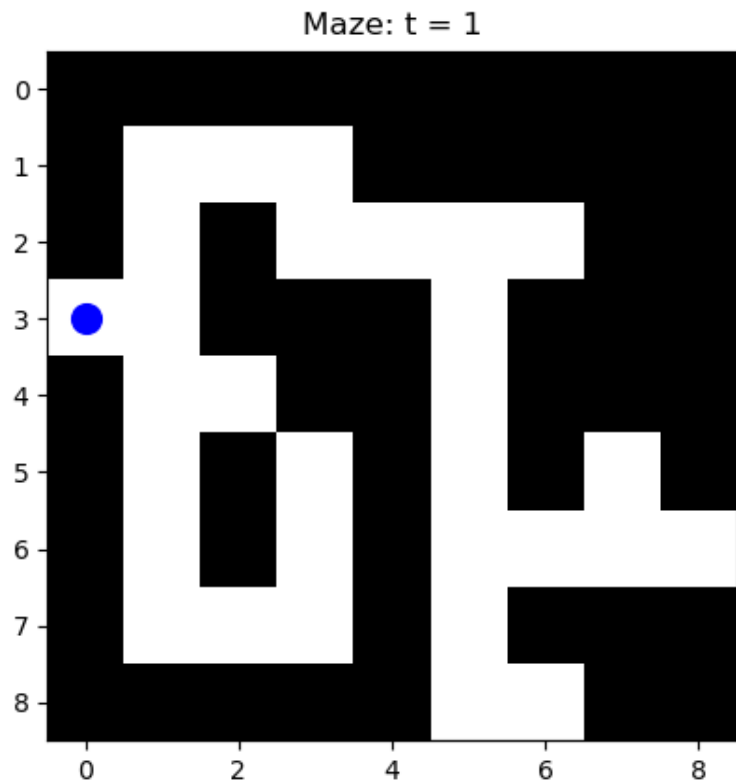
Fim

Objetivo: Chegar ao final do labirinto a partir da posição inicial.

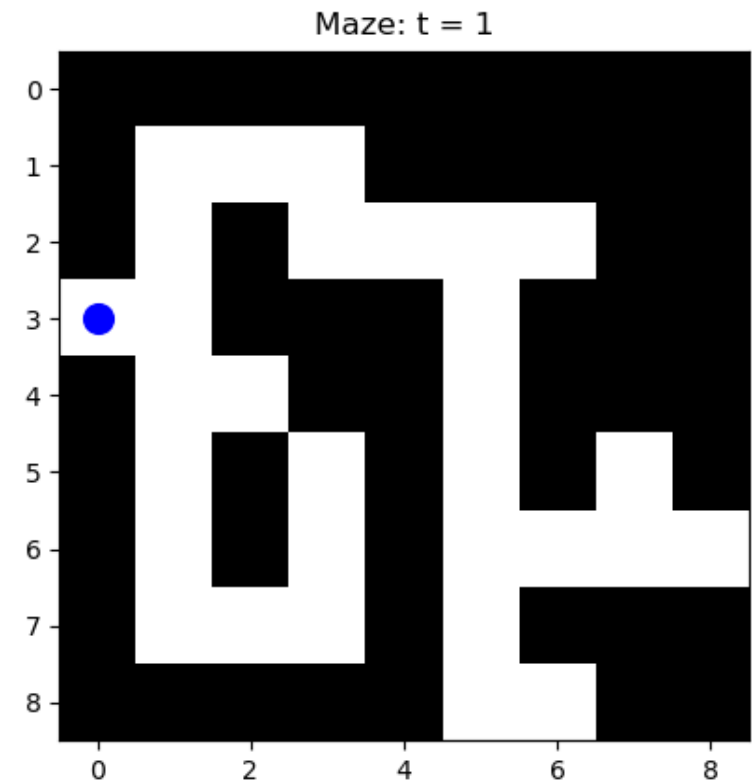
- Estados: Posições em que o agente se encontra.
- Ações: 0, \uparrow , \downarrow , \rightarrow , \leftarrow
- Recompensas: $r = -1$ para todo estado com exceção do final.

EXEMPLO: LABIRINTO: POLÍTICAS

Política de ações aleatória π_{rand}

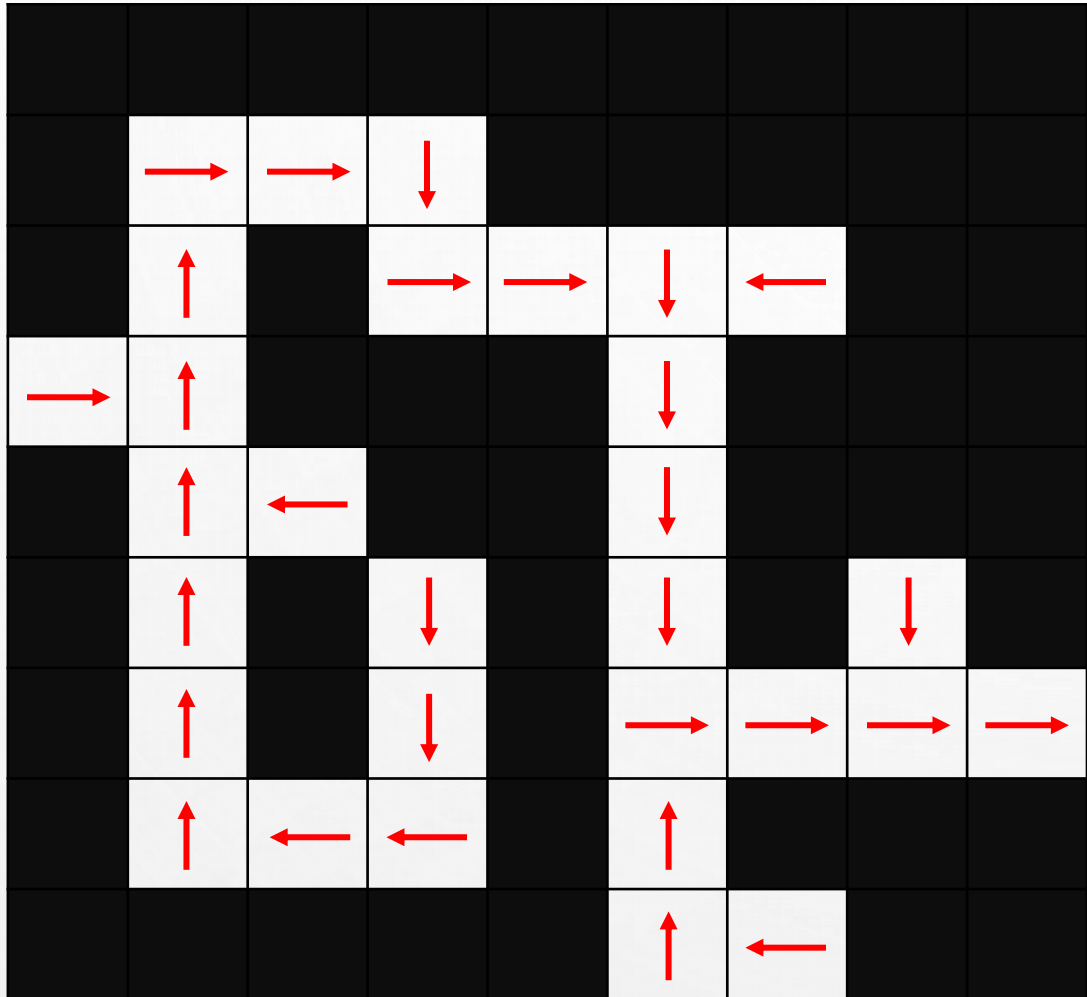


Política de ações ótima π^*



EXEMPLO: LABIRINTO: POLÍTICAS

Início

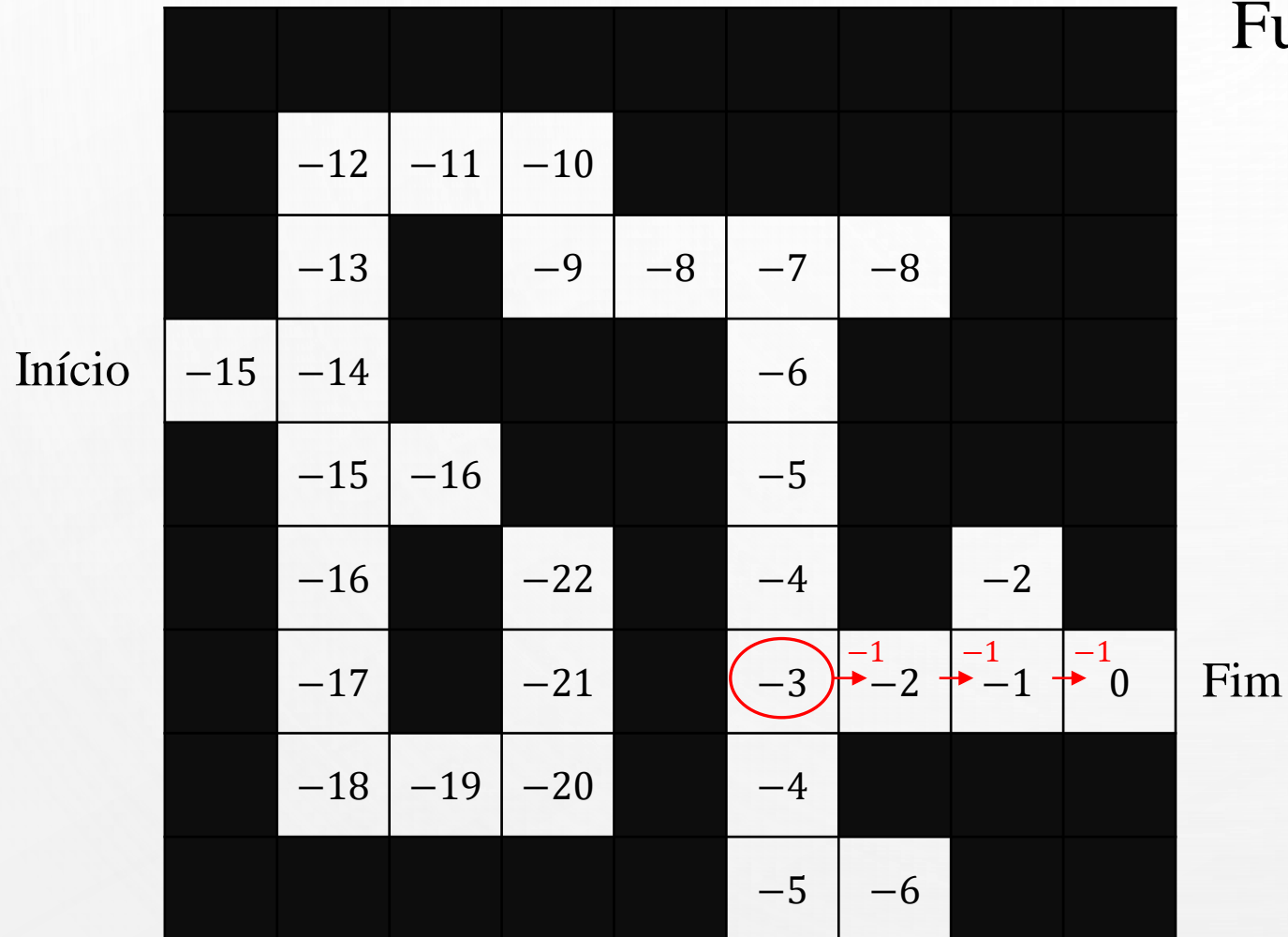


Fim

Política de ações ótima π^*

- $\pi^*([3,0]) = \pi^*([1,1]) = \pi^*([1,2]) = \pi^*([2,3]) = \pi^*([2,4]) = \pi^*([6,5]) = \pi^*([6,6]) = \pi^*([6,7]) = \pi^*([6,8]) = \rightarrow$
- $\pi^*([7,1]) = \pi^*([6,1]) = \pi^*([5,1]) = \pi^*([4,1]) = \pi^*([3,1]) = \pi^*([2,1]) = \pi^*([7,5]) = \pi^*([8,5]) = \uparrow$
- $\pi^*([7,2]) = \pi^*([4,2]) = \pi^*([8,6]) = \pi^*([2,6]) = \leftarrow$
- $\pi^*([5,3]) = \pi^*([6,3]) = \pi^*([2,5]) = \pi^*([3,5]) = \pi^*([4,5]) = \pi^*([5,5]) = \pi^*([5,7]) = \downarrow$

EXEMPLO: LABIRINTO: FUNÇÃO VALOR



Função Valor $V^*(s)$
 $(\gamma = 1)$

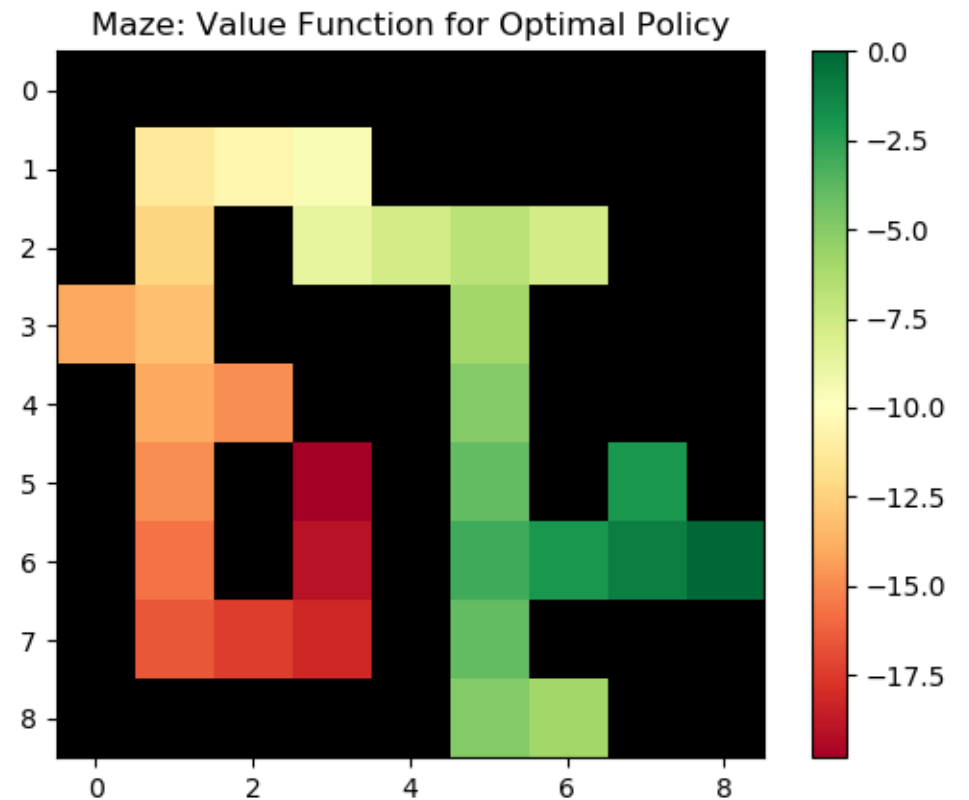
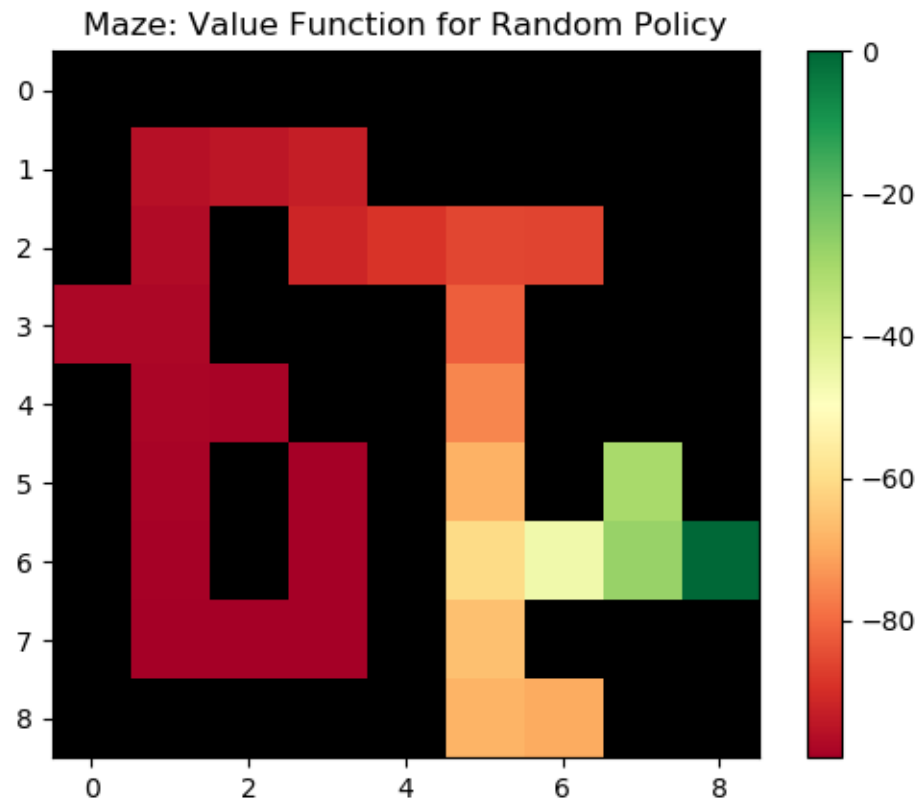
- $V^*([6, 5]) = -3 \Rightarrow$ Espera-se que, seguindo uma política ótima π^* a partir do estado $[6, 5]$ o agente receba uma recompensa acumulada de -3.

EXEMPLO: LABIRINTO: FUNÇÃO VALOR

Função Valor $V(s)$

$V_{\pi_{rand}}$ (Agente aleatório) ($\gamma = 0.99$)

V^* (Agente Ótimo)



\neq
$$M = \{\mathcal{P}_{SS'}^a, \mathcal{R}_S^a\}$$

	-1	-1	-1					
	-1		-1	-1	-1	-1		
-1	-1				-1			
	-1	-1			-1			
	-1		-1		-1		-1	
	-1		-1		-1	-1	-1	0
	-1	-1	-1		-1			
					-1	-1		

Fim

Início

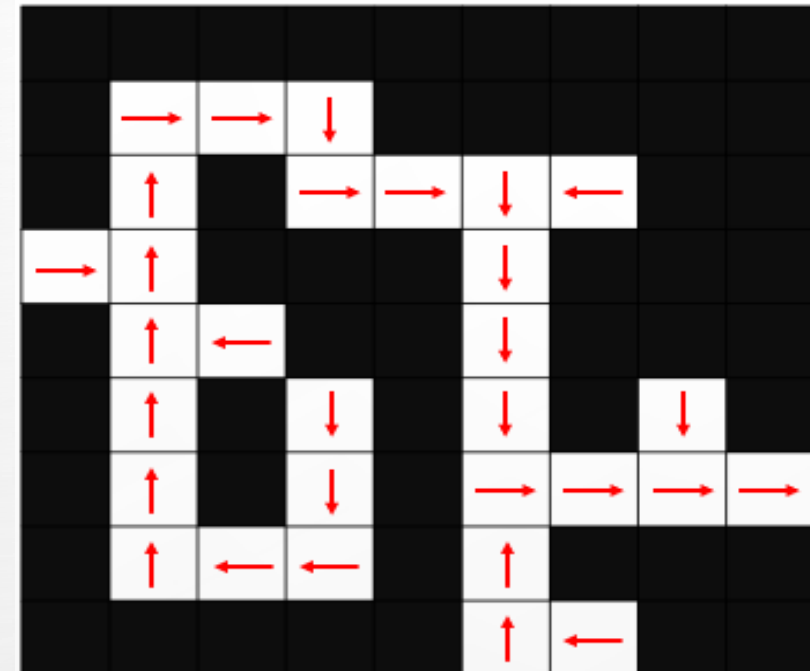
[illegible]

Fim

- Dada uma política de ações $\pi(s, a)$:
 - Determinar a função valor $V_\pi(s)$
 - Estudar probabilidades de sequências de estados.
- Exemplo: Qual é a função valor associada à política aleatória?

	-95.3	-93.9	-92.3					
	-96.4		-90.4	-87.4	-84.2	-84.7		
-97.3	-97.2				-79.3			
	-97.8	-97.8			-74.2			
	-98.2		-99.2		-67.7		-29.5	
	-98.4		-99.2		-60.0	-44.9	-27.2	0
	-98.7	-98.9	-99.1		-66.3			
					-70.1	-71.1		

- Determinar a função valor ótima $V^*(s)$ sobre todas possíveis políticas.
- Determinar a política de ações ótima $\pi^*(s, a)$.
- Exemplo: Qual é a política de ações ótima?



APRENDIZADO E PLANEJAMENTO

Aprendizado (Model-Free)

- O Ambiente é inicialmente desconhecido.
- Agente interage com o ambiente e melhora sua política de ações.

Planejamento (Model-Based)

- Modelo do ambiente é conhecido.
- Agente tem acesso ao modelo para realizar simulações sem a necessidade de interação externa.
- Agente melhora sua política de ações (DP, MCTS).

EXPLORATION-EXPLOITATION TRADE-OFF

Exploration (“Exploração”)

- Tomar ações (eventualmente sub-ótimas) com o objetivo de visitar novas regiões do espaço de estados e obter melhor conhecimento do ambiente.

Exploitation (“Aproveitamento”)

- Tomar ações ótimas dado o conhecimento atual com o objetivo de maximizar recompensas acumuladas.

Exploration-Exploitation Trade-Off:

- Política atual pode ser sub-ótima e ações não exploradas podem levar a maiores recompensas a longo prazo.
- Ao mesmo tempo, exploração pode demandar tempo e não levar a políticas melhores.
- Como conciliar as duas estratégias?

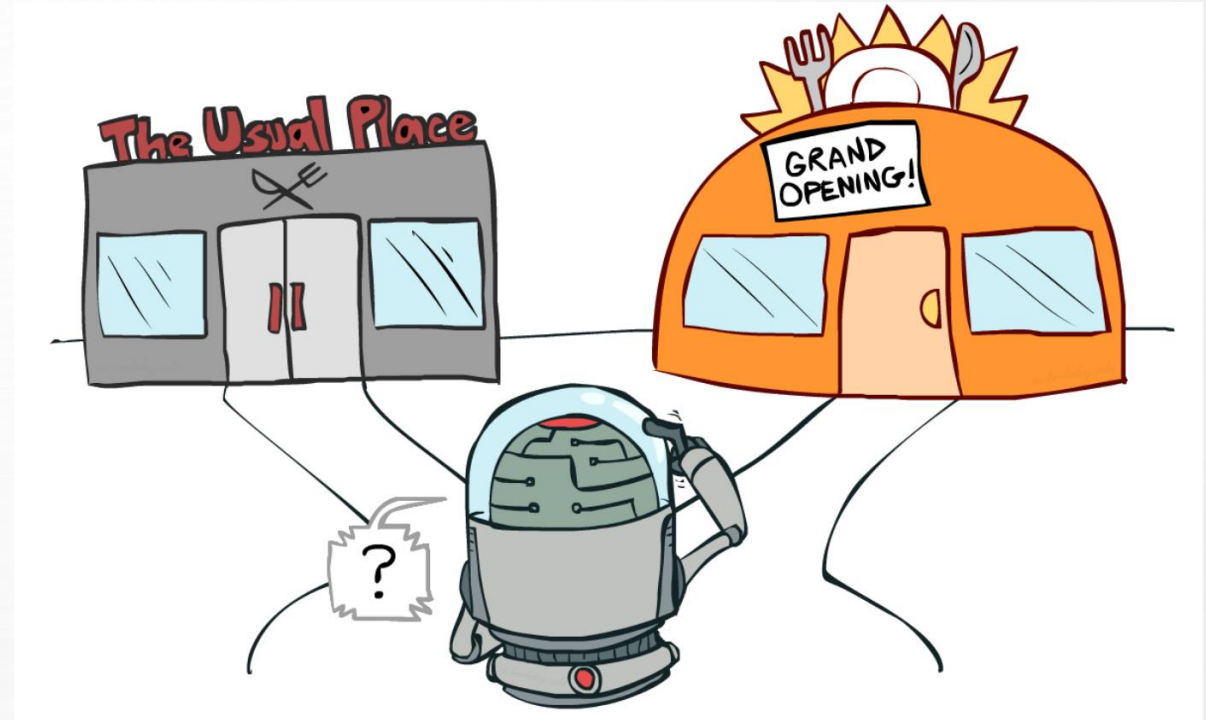
EXPLORATION-EXPLOITATION TRADE-OFF: EXEMPLOS

Escolher restaurante

- Exploration: Visitar novo restaurante.
- Exploitation: Ir no restaurante favorito.

Jogo de tabuleiro

- Exploration: Tentar movimento não convencional.
- Exploitation: Executar movimento que acredita ser melhor.



<https://medium.com/analytics-vidhya/the-epsilon-greedy-algorithm-for-reinforcement-learning-5fe6f96dc870>

EXERCÍCIO EXTRA: ALPHAGO

- Assistir documentário sobre algoritmo AlphaGo desenvolvido pelo DeepMind.
- Escrever approx. 1 página sobre o documentário destacando os pontos:
 - Utilidade do Aprendizado por Reforço em comparação com métodos de IA baseados em regras.
 - Importância dos resultados.
 - Estrutura do algoritmo.



<https://www.alphagomovie.com/>

Artigo:

Silver, D., Schrittwieser, J., Simonyan, K. *et al.* *Mastering the game of Go without human knowledge*, Nature 550, 354–359 (2017).

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] Sutton, R. and Barto, A. *Reinforcement Learning: An Introduction*, The MIT Press (2020).
- [2] Szepesvári, C. *Algorithms for Reinforcement Learning*, Morgan & Claypool Publishers (2009).
- [3] Silver, D., Schrittwieser, J., Simonyan, K. *et al.* *Mastering the game of Go without human knowledge*, Nature 550, 354–359 (2017).

Muito obrigado a todos!

Dúvidas